

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

LARISSA SILVA BAQUIÃO

MODELAGEM DE UM MOTOR SÍNCRONO COM ROTOR EXTERNO UTILIZANDO
O SOFTWARE ELMER

Joinville
2025

LARISSA SILVA BAQUIÃO

MODELAGEM DE UM MOTOR SÍNCRONO COM ROTOR EXTERNO UTILIZANDO
O SOFTWARE ELMER

Trabalho apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica, no Curso de Engenharia Mecatrônica, no Centro Tecnológico de Joinville, da Universidade Federal de Santa Catarina.

Orientadora: Dra. Aline Durrer Patelli Juli-
ani

Joinville
2025

Dedico este trabalho, primeiramente, a Deus. Estendo minha gratidão aos meus pais, ao meu namorado e a todos os meus familiares, pilares fundamentais desta jornada. E por fim, agradeço igualmente aos meus professores e amigos, cujo apoio foi indispensável.

AGRADECIMENTOS

A Deus, minha eterna gratidão por me sustentar ao longo desta jornada, por ser meu refúgio nas angústias e a razão para persistir quando a vontade era desistir.

À minha amada família, Luiza e Ivo, que me apoiou incondicionalmente em cada passo. Obrigada por investirem em meus sonhos com tudo o que tinham, tanto no suporte emocional quanto no financeiro.

Ao meu namorado, Alberto, por seu apoio constante, seu carinho que tornou os momentos felizes mais brilhantes e os difíceis, mais leves.

Às minhas tias, Eunice e Luciana, por suas orações que me protegeram todos os dias e por suas palavras de ânimo ao longo de todos esses anos. Um agradecimento especial aos meus avós, Lurdinha, Zé Baquião e Mariana, cujas histórias divertidas nos fins de semana eram um bálsamo.

Aos meus mestres e professores, pela paciência e dedicação ao transmitir seus conhecimentos e pela ajuda fundamental com exercícios que pareciam de outro mundo. Em especial, à minha orientadora, Aline, pela confiança em meu potencial, pela oportunidade e por todo o suporte oferecido.

Aos meus amigos, meu muito obrigada. O humor duvidoso e o carinho de vocês foram meu refúgio e motivo de muitas risadas nos cafés da cantina e nos almoços que celebravam o fim de cada prova e aula exaustiva.

Tenho a certeza de que esta conquista não seria possível sem cada uma dessas pessoas. Vocês são e sempre serão minha maior inspiração.

A todos, o meu muito obrigada.

“Se pude enxergar mais longe, foi porque me apoiei em ombros de gigantes.”

Isaac Newton

RESUMO

Um percentual elevado de toda energia elétrica consumida está relacionado com a aplicação de motores elétricos, principalmente, nas indústrias. Na etapa de projeto, uma das maneiras em se obter um motor eficiente é realizar seu dimensionamento e observar o desempenho teórico, utilizando um software baseado no método dos elementos finitos. Atualmente, o software livre FEMM é empregado para análises estáticas e em duas dimensões nas máquinas elétricas. No entanto, a necessidade de executar cálculos precisos das perdas, justificou-se análises no domínio do tempo. De acordo com essa necessidade, este trabalho apresenta a modelagem física e matemática de um motor rotativo síncrono com ímãs permanentes e rotor externo em duas dimensões, utilizando o software Elmer, que é empregado em problemas multifísicos e publicado em código aberto. As atividades realizadas envolveram a aprendizagem da utilização dos softwares Gmsh (pré-processamento), Elmer (processamento) e Paraview (pós-processamento), executando os cálculos das grandezas elétricas e magnéticas de um motor rotativo de 250-500 W, aplicado à tração elétrica.

Palavra-chave: software Elmer; motores elétricos; máquinas com ímã permanente;

ABSTRACT

A high percentage of all electricity consumed is related to the application of electric motors, especially in industries. During the design stage, one way to obtain an efficient motor is to perform its sizing and observe its theoretical performance using a software based on the finite element method. Currently, free software FEMM is used for static and two-dimensional analysis of electric machines. However, the need to perform precise calculations of losses requires time-domain analysis. To address this need, this work presents the physical and mathematical modeling of a two-dimensional rotating motor using the software Elmer, which is employed in multi-physics problems and published as open-source code. The activities performed included using Gmsh software for pre-processing, Elmer software for processing, and Paraview software for post-processing, executing the calculations of the electrical and magnetic magnitudes of a 250-500 W rotating motor, applied to electric traction.

Keywords: software Elmer; electric motor; machines with permanent magnets;

LISTA DE FIGURAS

Figura 1 – Motor Corrente Contínua Elementar	14
Figura 2 – Forma de onda do fluxo concatenado e tensão induzida	16
Figura 3 – Esquemático do acionamento elétrico	17
Figura 4 – Inversor trifásico e motor	18
Figura 5 – Gráfico das tensões de saída por fase em relação a posição do rotor	18
Figura 6 – Motor com Rotor Externo	19
Figura 7 – Malha triangular de elementos finitos	20
Figura 8 – Estrutura do rotor completa	25
Figura 9 – Tela inicial e script	25
Figura 10 – Aba para definir ponto	26
Figura 11 – Os 20 pontos definidos	28
Figura 12 – Desenhando linhas e conectando os pontos	29
Figura 13 – Desenhando elipses	29
Figura 14 – Desenhando semi círculos	30
Figura 15 – Estrutura primária	30
Figura 16 – Pontos iniciais para geometria do rotor	32
Figura 17 – Conectando os pontos com semicírculos	33
Figura 18 – Conectando os pontos com linhas	33
Figura 19 – Estrutura primária do rotor	34
Figura 20 – Entreferro	35
Figura 21 – Malha pronta na estrutura	39
Figura 22 – Cálculo pronto feito pelo Elmer, exemplo no prompt	41
Figura 23 – Print da pasta results que aparece na pasta de destino	41
Figura 24 – Print da pasta results que aparece na pasta de destino	42
Figura 25 – Motor Inicialmente com cor sólida.	43
Figura 26 – Motor com mostrando a densidade de fluxo magnético.	43
Figura 27 – Primeiro caminho de fluxo e configuração para criar-se mais caminhos	43
Figura 28 – Resultado das configurações anteriores	44
Figura 29 – Indução magnética no enteferro	45
Figura 30 – Gráfico do torque de Borda	46
Figura 31 – Configuração - Passo 1	47
Figura 32 – Configuração da Fase positiva - Passo 2	48
Figura 33 – Configuração da Fase positiva - Passo 3	49
Figura 34 – Cálculo da Psiu	50
Figura 35 – Configuração para salvar o arquivo em uma tabela	51
Figura 36 – Dimensões do motor com Rotor externo	53

Figura 37 – Fluxo concatenado em uma fase no cálculo da tensão induzida . . .	56
Figura 38 – Tensão induzida em uma fase (Elmer)	57
Figura 39 – Formas de onda das correntes de alimentação (fases A, B e C) . . .	58
Figura 40 – Torque Eletromagnético	58
Figura 41 – Comparação do fluxo magnético concatenado: Elmer vs. FEMM . .	59
Figura 42 – Comparação da tensão induzida: Elmer vs. FEMM	60
Figura 43 – Alimentação FEMM	61
Figura 44 – Comparação do torque eletromagnético sob carga: Elmer vs. FEMM	62

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	12
1.1.1	Objetivo geral	12
1.1.2	Objetivos específicos	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	MOTORES SÍNCRONOS COM ÍMÃS PERMANENTES	13
2.1.1	Funcionamento do motor síncrono com ímãs permanentes	13
2.1.1.1	Torque de Borda	16
2.1.1.2	Acionamento do motor BLDC	17
2.1.1.3	Inversor Trifásico	17
2.1.2	Motor com rotor externo	18
2.2	MÉTODO DE ELEMENTOS FINITOS	19
3	METODOLOGIA	24
3.1	SOFTWARE GMSH	24
3.1.1	Geometria Estator	26
3.1.2	Geometria do Rotor	31
3.1.3	Entreferro	35
3.1.4	Definição das Regiões Físicas	36
3.1.5	Geração da Malha no Gmsh	38
3.2	SOFTWARE ELMER	40
3.3	SOFTWARE PARAVIEW	41
3.3.1	Linhas de fluxo	42
3.3.2	Indução magnética no entreferro	44
3.3.3	Torque de Borda	45
3.3.4	Tensão Induzida	46
3.3.4.1	Motor alimentado	51
4	RESULTADOS	53
4.1	DIMENSÕES DA MÁQUINA	53
4.2	VALIDAÇÃO DO MODELO: COMPARAÇÃO ENTRE ELMER E FEMM	54
4.2.1	Ferramenta de Referência: FEMM	55
4.2.2	Análise dos cálculos no Elmer	55
4.2.2.1	Caracterização das tensões induzidas (Sem Alimentação)	55
4.2.2.2	Análise sob Carga (Com Alimentação)	57
4.2.3	Análise Comparativa e Validação dos Resultados	59
5	CONCLUSÃO	63
	REFERÊNCIAS	64

APÊNDICE A – CÓDIGO DE CÁLCULO DO TORQUE DE BORDA	66
APÊNDICE B – CÓDIGO DE CÁLCULO DA TENSÃO INDUZIDA E FLUXO CONCATENADO	67
APÊNDICE C – ARQUIVO .SIF PARA MODELAGEM 2D DO MO- TOR - MOTOR SEM ALIMENTAÇÃO	69
APÊNDICE D – ARQUIVO .SIF PARA CÁLCULO DO MOTOR COM ALIMENTAÇÃO	83

1 INTRODUÇÃO

Na preocupação mundial com a diminuição do aquecimento global e dos gases de efeito estufa, as pesquisas têm apontado alternativas limpas de produção energética, mantendo a eficiência e custo acessível. Segundo Hicham *et al.* (2020), a eletrificação dos motores é uma alternativa para alcançar esse objetivo, uma vez que apresentam uma eficiência na conversão de energia de 86% a 90%, além da confiabilidade quanto à manutenção e robustez. Com isso, seu uso é amplo no setor de eletrodomésticos, automotivo, industrial e aeronáutico.

De acordo com Karki *et al.* (2020), a vantagem dos motores elétricos para os convencionais está na maior eficiência, menor emissão de gases poluentes durante o funcionamento por não fazer uso de combustíveis fósseis e apresentar menor ruído. Essas vantagens, tornam o mercado de máquinas elétricas atrativo e estima-se que esse segmento alcance um valor de 206,4 bilhões de dólares até 2029, crescendo 152,2 bilhões de dólares em 2024 (Marktsanmarket, 2020).

O motor elétrico largamente utilizado ao longo dos últimos 130 anos é o de indução trifásico com rotor gaiola de esquilo, inventado por Nicolas Tesla em 1890 e, apesar da aplicação em diversas áreas devido à simplicidade, vem sendo substituído por motores síncronos com ímãs permanentes (Permanent Magnet Synchronous Motor) que estão presentes na maioria dos veículos elétricos puros e híbridos, devido à alta eficiência em velocidades variáveis e ao tamanho compacto. Basicamente, esse motor é composto por um rotor com ímãs permanentes, um estator, uma carcaça e controlador eletrônico, além de duas configurações possíveis, a com rotor interno e rotor na superfície, também chamado de externo (Rimpas *et al.*, 2023). Devido à versatilidade, facilidade de instalação, confiabilidade e alto torque, tem sido aplicado em áreas como aviação, transporte, automóveis e automação industrial.

Segundo Dornelles (2020), o desenvolvimento de motores elétricos modernos, como os de ímãs permanentes, permite a otimização do desempenho nas aplicações de engenharias em geral, pois reduz as perdas de energia e aumenta a confiabilidade operacional. No entanto, conforme Hendershot e Miller (2010), projetar um motor de ímãs permanentes não é um trabalho trivial, pois exige um conhecimento profundo de magnetismo, mecânica, termodinâmica, eletrônica, acústica e ciência dos materiais. Dessa forma, mesmo que a demanda seja crescente por esse tipo de motor, um estudo estruturado é necessário com aspectos como desempenho e eficiência.

Tendo isso em vista, apresenta-se neste trabalho a modelagem de um motor síncrono com ímãs permanentes e rotor externo, utilizando-se o software Elmer como ferramenta de cálculo para obtenção das grandezas elétricas e magnéticas por meio do método de elementos finitos. Além disso, a metodologia usada foi baseada no cál-

culo computacional, onde foi exequível desenhar a geometria e testá-la em diferentes condições. Nas subseções seguintes são apresentados os objetivos do trabalho.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Modelar um motor síncrono com ímãs permanentes, por meio do software Elmer.

1.1.2 Objetivos específicos

- Revisão das teorias sobre motores síncronos com ímãs permanentes e métodos dos elementos finitos;
- Realizar a modelagem do motor, abordando detalhadamente, o pré-processamento, processamento e pós-processamento. Em cada etapa, a metodologia utilizada deve ser explicada didaticamente, para que possa ser utilizada em trabalhos futuros;
- Comparar resultados obtidos no Elmer, como tensões induzidas e torque eletromagnético, com cálculos realizados no software FEMM.

2 FUNDAMENTAÇÃO TEÓRICA

Partindo da definição e operação de motores elétricos síncronos com ímãs permanentes, o cálculo computacional tem como ponto de partida o desenvolvimento matemático, utilizado para modelar o sistema em sua totalidade. Para isso, torna-se essencial o entendimento teórico do funcionamento de motores elétricos, bem como do método de modelagem de sua estrutura por meio do método de elementos finitos, que fundamenta a modelagem física. Neste capítulo são apresentados os conceitos matemáticos, o princípio de funcionamento e as técnicas de cálculo envolvidas na execução da modelagem. partindo da definição e operação de motores elétricos síncronos com ímãs permanentes.

2.1 MOTORES SÍNCRONOS COM ÍMÃS PERMANENTES

A Máquina Síncrona com Ímã Permanente, consiste em substituir o enrolamento de campo do rotor por ímãs permanentes, eliminando a necessidade de uma fonte externa de excitação no rotor, com isso, há uma redução das perdas, o que possibilita uma diminuição do volume e uma maior eficiência energética do sistema (Tavares, 2018).

Um motor síncrono com ímãs permanentes pode ser dividido em duas categorias a depender de sua alimentação. Quando é senoidal a máquina síncrona é chamada de Brushless AC (BLAC) e quando sua alimentação é trapezoidal, é denominada de Brushless DC (BLDC) (Juliani, 2024).

Motores síncronos apresentam uma velocidade de rotação diretamente proporcional à frequência de alimentação elétrica e inversamente proporcional ao número de pares de polos do sistema (Hendershot, 2010). Tendo isso em vista, essa relação é descrita pela Equação 1.

$$N = \frac{f \times 60}{p}, \quad (1)$$

Onde N corresponde à velocidade síncrona (rpm), f representa a frequência (Hz) de alimentação e p indica o número de pares de polos. Na sequência é abordado com mais detalhes o funcionamento e nomenclatura para esse tipo de motor.

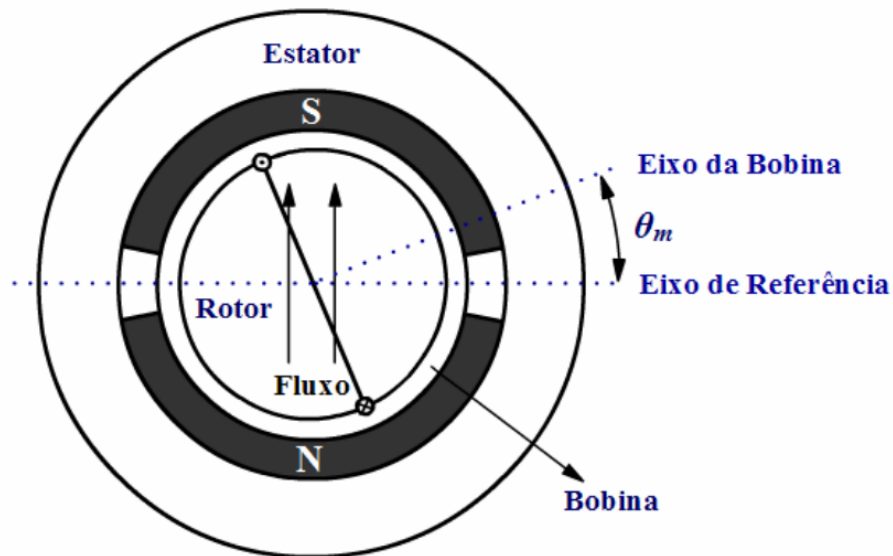
2.1.1 Funcionamento do motor síncrono com ímãs permanentes

A denominação motor brushless (sem escovas) refere-se à ausência de escovas, anéis deslizantes e comutador mecânico. Essa nomenclatura origina-se do fato

de essa topologia de motor apresentar operação semelhante à de um motor CC com escovas. A principal diferença consiste na substituição do enrolamento de campo por ímãs permanentes e no uso de um comutador eletrônico, implementado por meio de conversores eletrônicos de potência (Hendershot, 2010).

O comportamento do motor BLDC em conjunto com seu acionamento assemelha-se ao de máquinas CC tradicionais. A Figura 1 apresenta um motor CC elementar que permite compreender o funcionamento. De forma simplificada, a rotação do rotor e a interação do campo magnético gerado pelos ímãs permanentes induzem a tensão na bobina (Hendershot, 1994).

Figura 1 – Motor Corrente Contínua Elementar



Fonte: Juliani (2007, p. 17).

Analisando com maior detalhamento a Figura 1, observa-se o deslocamento angular do rotor θ_m em graus mecânicos, bem como o fluxo magnético produzido pelos ímãs permanentes. Ademais, quando o ângulo θ_m é igual a zero, não ocorre passagem de fluxo magnético pela espira, resultando em fluxo concatenado ψ nulo, o mesmo acontece quando θ_m alcança 180° . O fluxo magnético atinge sua máxima amplitude e se mantém constante por um intervalo de tempo, devido ao espaço vazio que existe entre os polos (Hendershot, 1994).

De acordo com Ponomarev (2015), o acoplamento de fluxo ψ pode ser determinado a partir do vetor potencial magnético A , por meio da integração sobre as superfícies correspondentes aos terminais positivo e negativo da fase, conforme apresentado na Equação 2.

$$\psi(\theta_m) = N_{ph} l_{stack} \left(\int_{S_{phase+}} A dS - \int_{S_{phase-}} A dS \right) \quad (2)$$

onde:

- N_{ph} é o número de espiras por fase;
- l_{stack} é o comprimento axial do pacote do estator (m);
- A é o vetor potencial magnético (Wb/m);
- S_{phase+} e S_{phase-} representam, respectivamente, as superfícies positiva e negativa da fase.

A partir da Lei de Faraday, a tensão induzida nas bobinas pode ser obtida derivando-se o fluxo concatenado em relação ao tempo, conforme a Equação 3 (Hendershot, 1994):

$$e = -N \frac{d\psi}{dt} \quad (3)$$

Considerando que a velocidade angular mecânica é dada por $\omega_m = \frac{d\theta_m}{dt}$, a Equação 3 pode ser reescrita em função da posição angular mecânica θ_m , resultando na Equação 4:

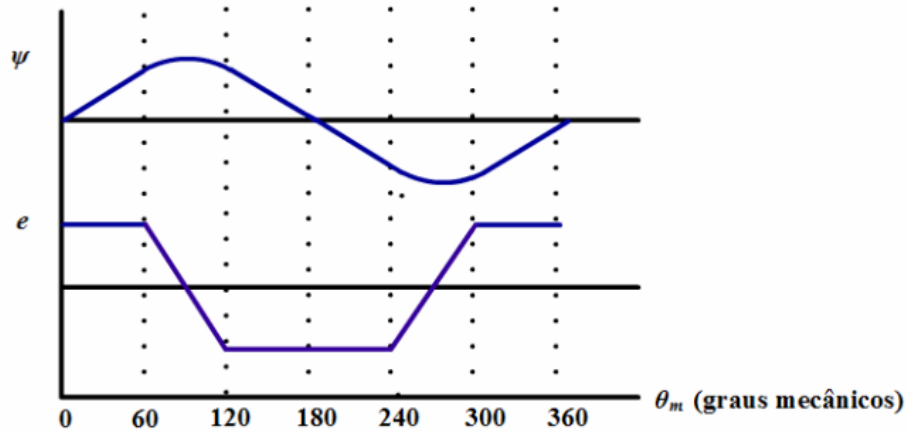
$$e = -N \frac{\partial \psi}{\partial \theta_m} \frac{d\theta_m}{dt} = -N \omega_m \frac{\partial \psi}{\partial \theta_m} \quad (4)$$

As variações do fluxo concatenado ψ em função do ângulo mecânico podem ser observadas na Figura 2. O torque eletromagnético T_{el} (N·m) correspondente é calculado pela Equação 5, a partir das componentes de fluxo e corrente de cada fase.

$$T_{el} = \frac{e_a \cdot i_a + e_b \cdot i_b + e_c \cdot i_c}{\omega_m} \quad (5)$$

Onde i é a corrente elétrica.

Figura 2 – Forma de onda do fluxo concatenado e tensão induzida



Fonte: Juliani, (2007, p. 18).

2.1.1.1 Torque de Borda

O torque de borda (cogging torque) consiste em uma parcela de torque indesejada que se manifesta em motores elétricos em decorrência da interação magnética entre os ímãs permanentes do rotor e as ranhuras do estator. Esse fenômeno ocorre independentemente da excitação elétrica nos enrolamentos, estando portanto, intrinsecamente associado às características construtivas e ao comportamento eletromagnético das máquinas elétricas (Paula, 2013).

Neste trabalho, utilizou-se como base o cálculo numérico do torque via Tensor de Maxwell. De maneira geral, o torque de borda é feito a partir da integração dos componentes da densidade de fluxo magnético ao longo de um caminho fechado (l_g) no entreferro do motor (Bianchi, 2005), assim, a equação contínua é dada na Equação 6.

$$T = p \frac{L_{Fe} D}{2\mu_0} \int_{l_g} B_r B_\theta dl \quad (6)$$

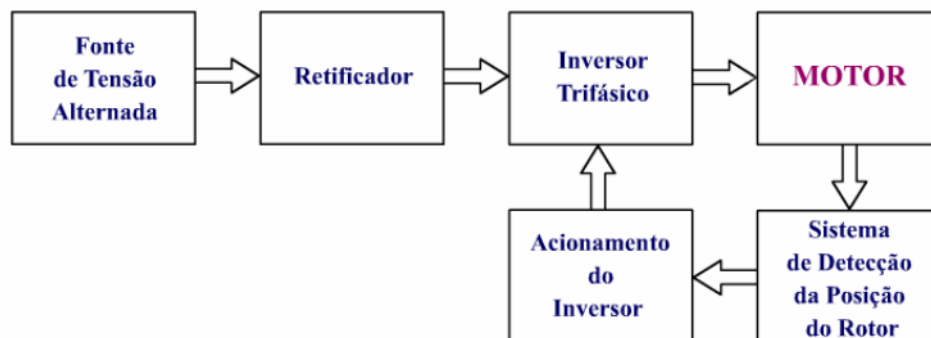
Onde p é o número de pares de polos, L_{Fe} é o comprimento axial do motor, D é o diâmetro do caminho de integração, μ_0 é a permeabilidade do vácuo, e B_r e B_θ são as componentes radial e tangencial da densidade de fluxo magnético, respectivamente. A partir desses parâmetros, o software de elementos finitos (FEM) aproxima esta integral de forma numérica através de um processo de discretização, tornando possível realizar os cálculos propostos nos objetivos específicos.

Após compreender como a rotação do rotor gera um fluxo e faz com que o motor funcione, é necessário explorar de que maneira ele é acionado, ou seja, como é dada sua partida.

2.1.1.2 Acionamento do motor BLDC

Majoritariamente, o acionamento elétrico de um motor síncrono com ímãs permanentes é composto por: uma fonte de tensão, um retificador, um conversor trifásico, sistema de detecção de posição do rotor (sensores) e acionamento do inversor. Na Figura 3, pode-se acompanhar a lógica do funcionamento do acionamento por meio do esquemático (Juliani, 2007).

Figura 3 – Esquemático do acionamento elétrico



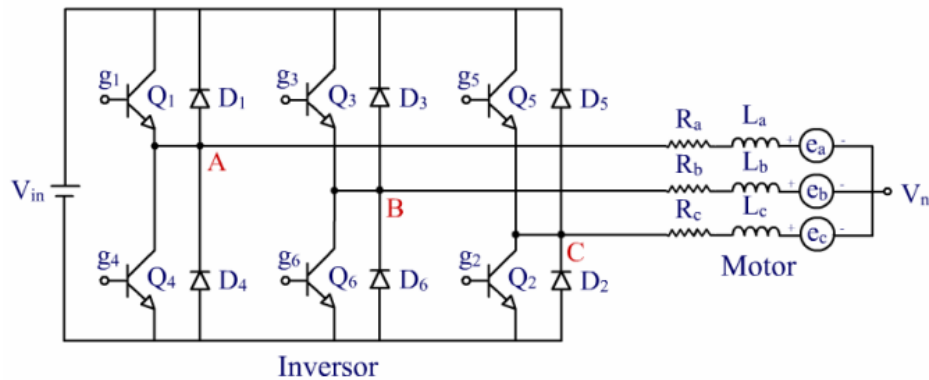
Fonte: Juliani (2007, p. 18).

2.1.1.3 Inversor Trifásico

Inversores são circuitos eletrônicos de potência, são utilizados em diversas aplicações para gerar tensões alternadas (CA) a partir da tensão retificada (CC), como fontes de alimentação ininterrupta (UPS), acionamento de motores e filtros ativos [Pineiro et al. 2005]. Neste contexto, os inversores trifásicos são aplicados ao acionamento de motores (Santos, 2011).

Pode-se observar na Figura 4, que o inversor trifásico consiste em três inversores monofásicos conectados em paralelo, onde em cada etapa de operação dois transistores, um superior e outro inferior, estarão conduzindo durante um intervalo de tempo (Fitzgerald, 2006).

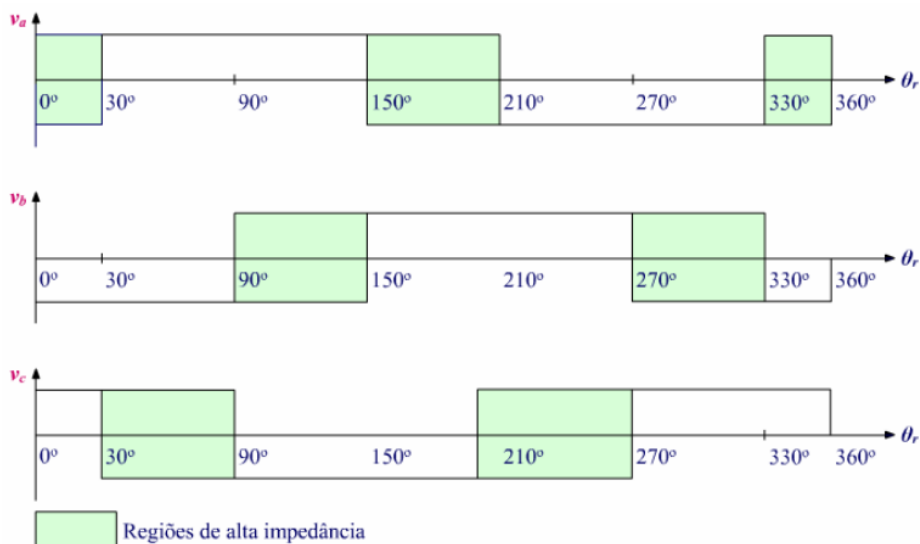
Figura 4 – Inversor trifásico e motor



Fonte: Juliani (2024, p. 268).

Nesse cenário, os transistores de potência recebem comandos da malha de controle do sistema, sincronizando-os com a posição do rotor (θ_m). A partir disso é possível determinar a tensão de cada fase A, B, C (Hendershot, 2010). Na Figura 5, é possível visualizar o comportamento das tensões de saída da fase A, B e C (v_a, v_b, v_c). A partir do conhecimento geral sobre motores elétricos com ímãs permanentes, é importante especificar ainda mais o tipo de motor alvo desse trabalho. Dessa forma, o estudo continua, agora, analisando os motores elétricos com ímãs permanentes e rotor externo.

Figura 5 – Gráfico das tensões de saída por fase em relação a posição do rotor



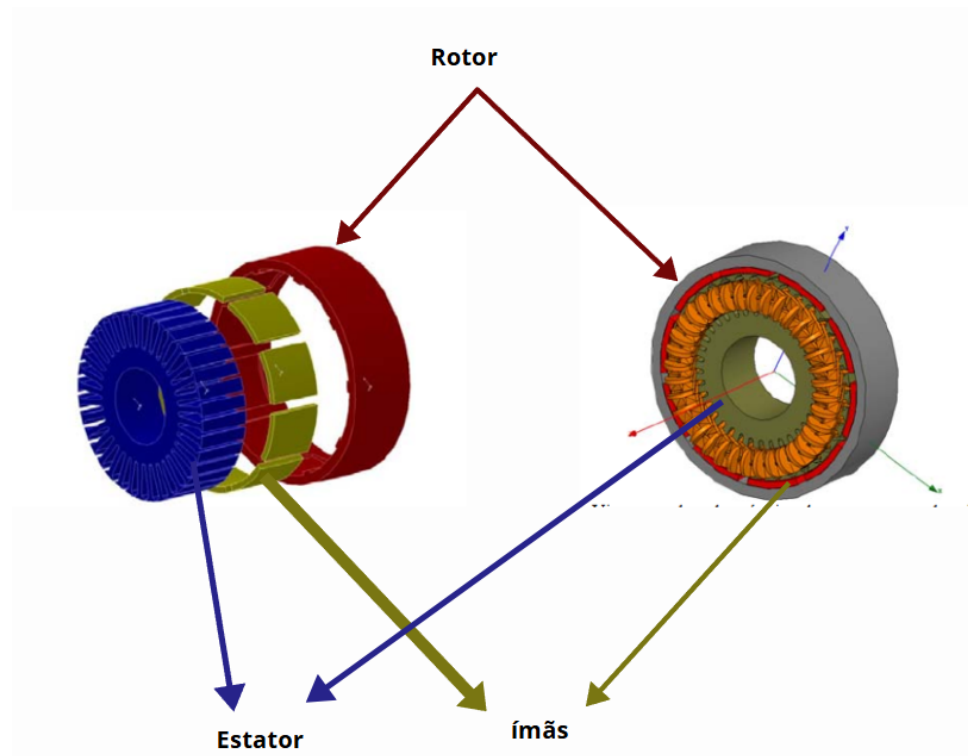
Fonte: Juliani (2007, p. 20).

2.1.2 Motor com rotor externo

Os motores brushless com rotor externo são uma solução adequada para aplicações que exigem alta rotação e velocidade constante, uma vez que apresenta baixo

custo de produção e um processo de fabricação relativamente simples, como visto na Figura 6. A inércia significativa do rotor é vantajosa em aplicações como ventiladores e sopradores. Além disso, a disposição dos ímãs no rotor não é um aspecto crítico, pois as forças que atuam sobre eles durante o funcionamento não favorecem seus deslocamentos (Hendershot e Miller, 1994).

Figura 6 – Motor com Rotor Externo



Fonte: Adaptado Tarimer e Ocak (2009, p. 6).

Na engenharia, tanto o projeto quanto a escolha de um motor para uma determinada aplicação, são realizados com base em uma análise de critérios. Dentre esses critérios, o torque no eixo é fundamental, pois é necessário verificar se o equipamento será capaz de atender às demandas do produto (Tarimer; Ocak, 2009).

2.2 MÉTODO DE ELEMENTOS FINITOS

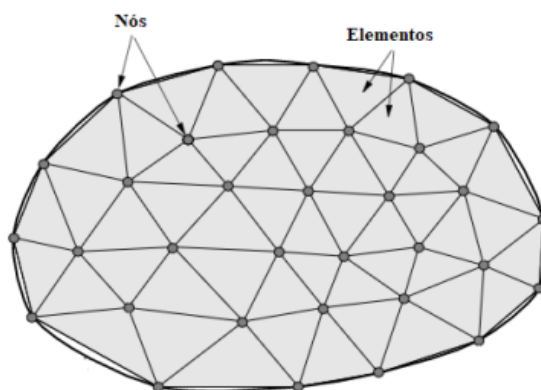
A crescente demanda por precisão no projeto e na análise das máquinas elétricas impulsionou o uso de modelos numéricos adequados. Com esses métodos, tem-se por objetivo determinar a distribuição dos potenciais magnéticos nas estruturas dessas máquinas, com base na resolução das equações de Maxwell (Hendershot, 1994). No entanto, devido à complexidade das geometrias das máquinas e à natureza não linear dos materiais, alcançar uma solução analítica é uma tarefa difícil e pouco recomendada (Fitzgerald, 2006).

Nesse contexto, o método dos elementos finitos (MEF) surge como uma técnica numérica eficaz, permitindo a obtenção de soluções para campos variáveis no tempo e para materiais não homogêneos, anisotrópicos ou não lineares. Nesse processo, o domínio de análise é subdividido em partes menores, conhecidas como elementos finitos, e as equações de campo são resolvidas individualmente para cada uma dessas partes. O MEF é organizado em cinco passos: particionamento do domínio, escolha das funções interpoladoras, formulação de um sistema para resolver o problema e por fim a solução do problema (Bianchi, 2005).

A primeira etapa do MEF consiste na subdivisão do domínio em regiões menores chamadas de elementos. Essa impacta diretamente a precisão da solução e o espaço de memória que será utilizado no computador. Em problemas unidimensionais, o domínio é uma curva composta por segmentos. Já em problemas bidimensionais, o domínio é uma superfície subdividida em polígonos, geralmente triângulos ou retângulos e em problemas tridimensionais, o domínio é um volume subdividido em tetraedros, prismas triangulares ou sólidos retangulares (Bastos, 2008).

Segundo Bohrer (2019), o conjunto de elementos em forma triângulos do domínio original é chamado de malha e os vértices são chamados de nós que estão relacionado ao grau de liberdade do elemento. Na Figura 7, é possível ver um exemplo de malha triangular.

Figura 7 – Malha triangular de elementos finitos



Fonte: Souza, 2003, p. 18.

A segunda etapa consiste na escolha de uma função de interpolação, que será responsável por aproximar as funções desconhecidas em cada elemento. Podem ser escolhidas funções de primeira, segunda ou terceira ordem. É importante observar que, quanto maior o grau da equação, mais complexos se tornam os cálculos. Por esse motivo, geralmente opta-se por equações de primeira ordem, ou seja, lineares (Juliani, 2007). Tendo definido o grau do polinômio escolhido, a solução desconhecida para cada n -ésimo elemento será dada pela Equação 7.

$$\phi_m^* = \sum_{j=1}^n \Phi_{mj} \nu_{mj}(x, y, z, t) \quad (7)$$

Onde as variáveis da função que aproxima ($\phi_m^* = \Phi_{mj}$) são:

- n é o número de nós do elemento.
- Φ_{mj} é o valor de ϕ no j -ésimo nó do m -ésimo elemento.
- ν_{mj} é a função interpoladora associada ao j -ésimo nó do m -ésimo elemento.

Para calcular o campo, os valores de Φ_{mj} precisam ser calculados nos nós de cada elemento. É necessário preparar um sistema de equações, cuja solução corresponde aos valores de Φ_{mj} . Para desenvolver esse sistema, tanto o método variacional (métodos de Rayleigh-Ritz) quanto o método dos resíduos (método de Galerkin) podem ser adotados, mas aplicados a cada elemento separadamente (Bianchi, 2005). Devido ao fato do software Elmer usar o métodos Galerkin (Ruokolainen et al., 2023), foi realizado o desenvolvimento do método Galerkin baseado na referência de Bianchi, (2005).

O Método dos Resíduos parte da suposição de que a função ϕ_m^* que melhor se aproxima da solução exata Φ_{mj} corresponde a um resíduo (r) r_m (Equação 8) igual a zero (ou muito baixo) em todo o domínio de análise.

$$r_m = L\phi_m^* - f_m \quad (8)$$

Onde L é o operador diferencial, ϕ_m^* é a solução aproximada (Equação 7) e f_m é o vetor de excitação no n -ésimo elemento.

Para garantir que o resíduo seja minimizado, o Método de Galerkin, em seguida, impõe que a integral do resíduo, ponderada pelas funções peso ν_{mi} (onde ν_{mi} é igual à função de interpolação, caracterizando o Método de Galerkin), seja nula sobre o volume do elemento τ , conforme equação 9.

$$r_i = \int_{\tau} \nu_{mi} r_m d\tau = 0 \quad (9)$$

Substituindo o resíduo r_m da Equação 8 na Equação 9, obtém-se a expressão da Equação 10:

$$r_i = \int_{\tau} \nu_{mi} (L\phi_m^* - f_m) d\tau = 0 \quad (10)$$

A partir da Equação 10, e utilizando a Equação 7, que define ϕ_m^* , o método dos resíduos chega à expressão das n integrais para o n -ésimo elemento, conforme a Equação 11:

$$r_i = \int_{\tau} \nu_{mi} L \left(\sum_{j=1}^n \Phi_{mj} \nu_{mj} \right) d\tau \quad (11)$$

Onde $i = 1, 2, \dots, n$.

Igualando a zero todos os termos da equação 11, um sistema de equações é obtido e expresso pela equação 12. Neste trabalho, o software Elmer resolve esse sistema linear pelo método direto, especificamente pelo Método MultiFrontal Assimétrico (UMFPACK, do inglês Unsymmetric MultiFrontal PACKage), conforme Ruokolainen (2023).

$$[SS][\Phi] = [T] \quad (12)$$

onde:

- $[SS]$ é a matriz de coeficientes, que depende das funções de interpolação.
- $[\Phi]$ é o vetor das incógnitas.
- $[T]$ é o vetor com o somatório das contribuições que não dependem da incógnita calculada.

Dessa forma, a fundamentação teórica aqui apresentada estabelece o embasamento conceitual e matemático para a modelagem e análise do motor síncrono de ímãs permanentes com rotor externo. Foram detalhados o princípio de operação da máquina, as relações entre velocidade síncrona, fluxo concatenado, tensão induzida e torque eletromagnético, bem como o cálculo do torque de borda.

Por fim, o detalhamento do método, incluindo o processo de discretização do domínio, a escolha das funções interpoladoras e a formulação do sistema de equações lineares via Método de Galerkin, finaliza o arcabouço teórico. Essa seção validou o procedimento numérico a ser empregado pelo software Elmer, demonstrando como a geometria complexa e a não linearidade dos materiais da máquina podem ser aproximadas de forma precisa para a obtenção das grandezas eletromagnéticas. A

partir desse referencial teórico-matemático, a próxima etapa consistiu em aplicar os conceitos aos cálculos do motor, cujos os métodos foram detalhados no capítulo 3.

3 METODOLOGIA

A metodologia de trabalho adotada neste estudo baseia-se em uma abordagem mista de pesquisa bibliográfica e cálculo computacional, sendo estruturada em três etapas principais do cálculo. O processo inicia-se com o pré-processamento no Gmsh, utilizado para a criação da geometria e geração da malha do motor síncrono. Em seguida, o processamento foi realizado no software Elmer, onde as equações de campo são resolvidas numericamente. Por fim, o pós-processamento e a visualização dos resultados foram executados no ParaView. Este capítulo detalha o fluxo dessas atividades computacionais, fundamentado na metodologia proposta por Panomarev (2015).

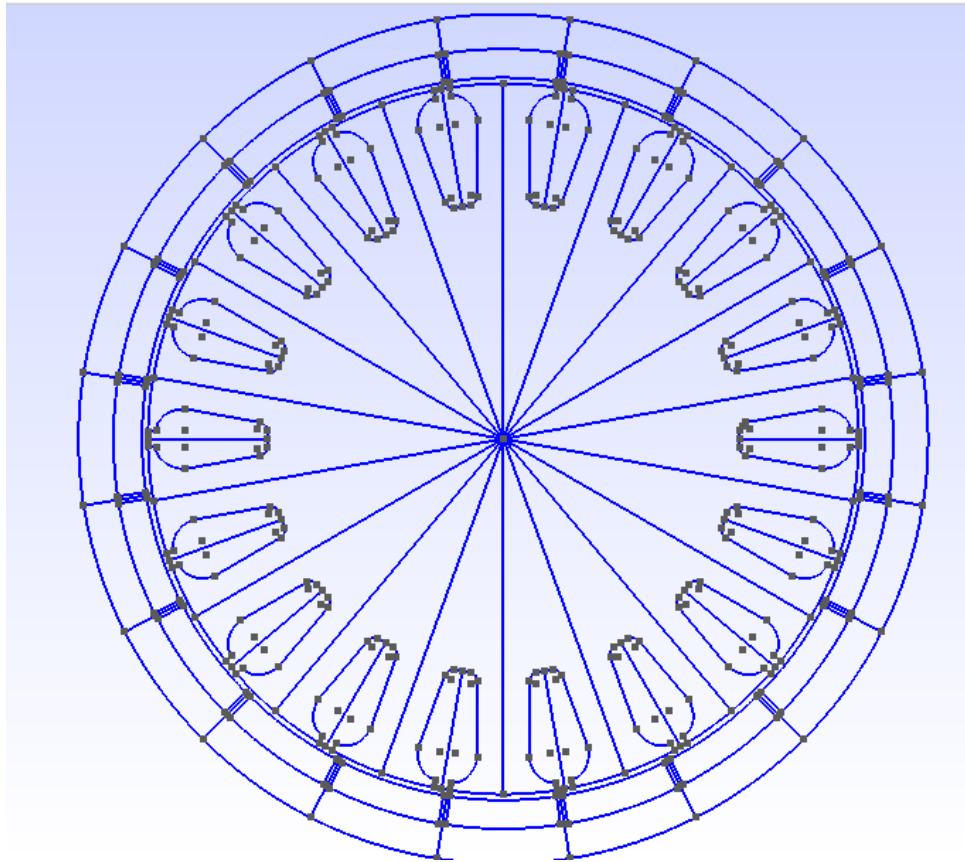
3.1 SOFTWARE GMSH

O software Gmsh é um gerador de malhas para elementos finitos em bi-dimensionais (2D) e tridimensionais (3D) com um mecanismo de Projeto Assistido por Computador que vem do inglês Computer-Aided Design (CAD) embutido e pós-processador. Pode ser executado no Windows, Linux e MacOS. Seu objetivo é fornecer uma ferramenta de malha eficiente e intuitiva, com entrada paramétrica e recursos avançados de visualização. O Gmsh é construído em torno de quatro módulos: geometria, malha, processamento e pós-processamento (Panomarev, 2015). Neste trabalho é usado para criar geometria e gerar as malhas.

Na sequência, as etapas básicas para a criação da geometria no Gmsh foram detalhadas. É possível utilizar a interface gráfica, baseando-se em linhas de comando (Geuzaine et al., 2009).

O propósito é alcançar a configuração geométrica completa do domínio de análise, a qual constitui o pré-requisito fundamental para a subsequente geração da malha computacional, conforme especificado na Figura 8.

Figura 8 – Estrutura do rotor completa

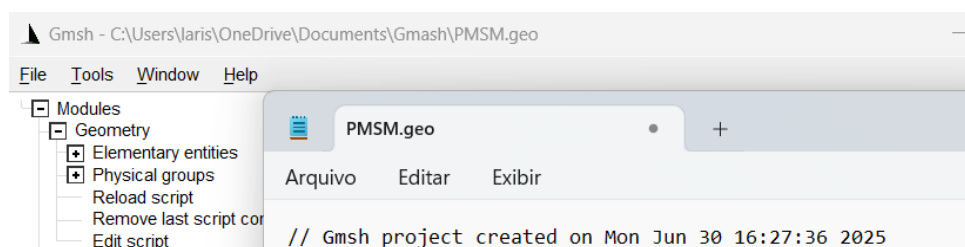


Fonte: Autora, 2025.

Previamente à definição paramétrica dos pontos no ambiente de software, é realizado por meio do levantamento dimensional da máquina física, empregando o sistema de coordenadas polares para determinar as distâncias e ângulos. Além disso, em alguns casos é necessário, desenhar o motor no papel utilizando esses pontos para ter clareza de onde posicioná-los no espaço.

No software *Gmsh*, seleciona-se a opção para criar um novo arquivo e nomeie-a com a extensão .geo, por exemplo PMSM.geo. Em seguida, o arquivo pode ser aberto em um editor de texto (como o Bloco de Notas), onde será escrito o script de modelagem, conforme Figura 9.

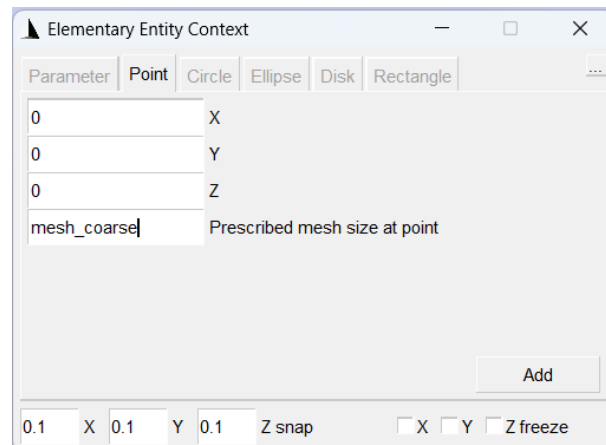
Figura 9 – Tela inicial e script



Fonte: Autora, 2025.

Para definir o ponto central da estrutura no *Gmsh*, o acesso é feito pelo menu Geometry → Elementary entities → Add → Point, conforme ilustrado na Figura 10. Como o objetivo é posicionar o ponto na origem do sistema de coordenadas, as coordenadas $\{0, 0, 0\}$, são inseridas. No campo Prescribed mesh size at point, o tamanho da malha desejada, utilizando o valor da variável `mesh_coarse`, é definida.

Figura 10 – Aba para definir ponto



Fonte: Autora, 2025.

É importante destacar que, embora seja possível definir os pontos manualmente pela interface gráfica do software, essa abordagem se torna imprecisa e trabalhosa à medida que o número de pontos aumenta. Isso se dá porque a exata posição de cada ponto depende diretamente das dimensões da máquina elétrica em questão. Além disso, os campos de entrada para coordenadas na interface gráfica do *Gmsh* aceitam apenas valores numéricos fixos, não permitindo o uso de expressões com variáveis ou constantes previamente definidas. Portanto, para garantir maior precisão e flexibilidade na modelagem, é preferível utilizar o script diretamente, onde é possível empregar variáveis e automatizar o posicionamento dos pontos.

3.1.1 Geometria Estator

Nesta etapa, é iniciada a construção da geometria do estator definindo-se os pontos que serviram de base para a geração da estrutura. Primeiro, estabeleceu-se o ponto central do estator. Em seguida, mais dois pontos foram definidos ao longo do eixo horizontal, correspondentes aos raios internos e externos da máquina.

A princípio, definiu-se a base geométrica de uma máquina elétrica no *Gmsh*, utilizando a biblioteca *OpenCASCADE* para modelagem. Primeiramente, foram definidos parâmetros constantes que representam as dimensões principais da máquina, como os raios interno e externo do rotor (R_{rot_in} , R_{rot_out}), o raio externo do estator (R_{stat_out}), o entreferro (Gap) e dimensões do dente, como a largura da base (b_d), a altura total da ranhura (h_{ss_tot}), a largura e altura do ímã (w_{PM} , h_{PM}). Em

seguida, foram definidos parâmetros relacionados à malha, como a densidade em regiões que necessitam maior precisão nos cálculos (`mesh_gap`, `mesh_fine`) e nas áreas menos críticas (`mesh_coarse`). O código abaixo mostra os parâmetros que foram definidos para essa máquina.

```

1 // Dimensoes da maquina
2 DefineConstant[ R_rot_in = { 45.5}];
3 DefineConstant[ R_rot_out = { 53.5}];
4 DefineConstant[ R_stat_out = { 44.75}];
5 DefineConstant[ Gap = { 0.75}];
6 DefineConstant[ b_d = { 21}];
7 DefineConstant[ hss_tot = { 32.4}];
8 DefineConstant[ bridge = { 0.6}];
9 DefineConstant[ w_PM = { 39}];
10 DefineConstant[ h_PM = { 3.6}];
11 DefineConstant[ h_wedge = { 3}];
12 // Pontos de densidade da malha
13 DefineConstant[ mesh_gap = {Gap/2}];
14 DefineConstant[ mesh_fine = { 0.5}];
15 DefineConstant[ mesh_normal = { 3}];
16 DefineConstant[ mesh_coarse = { 10}];
17 //=====
18 // Definicao da geometria
19 Point(1) = {0, 0, 0, mesh_coarse};

```

Listing 3.1 – Parâmetros da máquina elétrica

Após isso, iniciou-se a construção da geometria. O ponto 1 é o centro do estator, localizado na origem. Os pontos 2, 3 e 4 representam a base da ranhura (próxima ao centro), delimitando sua largura. Os pontos 5 a 8 delimitam a ranhura próxima ao entreferro (próxima ao centro). Já os pontos 9 a 12 delimitam a abertura da ranhura. Pontos adicionais, como 13 a 17 são referências centrais para os formatos circulares. A definição completa é exposta no código abaixo e pode ser verificada na Figura 11.

```

1
2
3 // =====
4 // ===== Geometria do Estator =====
5 // =====
6 //Dente do estator
7 Point(2) = {29.7, 0, 0, mesh_normal};
8 Point(3) = {29.7, 1.05, 0, mesh_normal};
9 Point(4) = {29.7, -1.05, 0, mesh_normal};
10 Point(5) = {30.56, 2.08, 0, mesh_normal};
11 Point(6) = {40.09, 3.78, 0, mesh_normal};
12 Point(7) = {30.56, -2.08, 0, mesh_normal};
13 Point(8) = {40.09, -3.78, 0, mesh_normal};

```

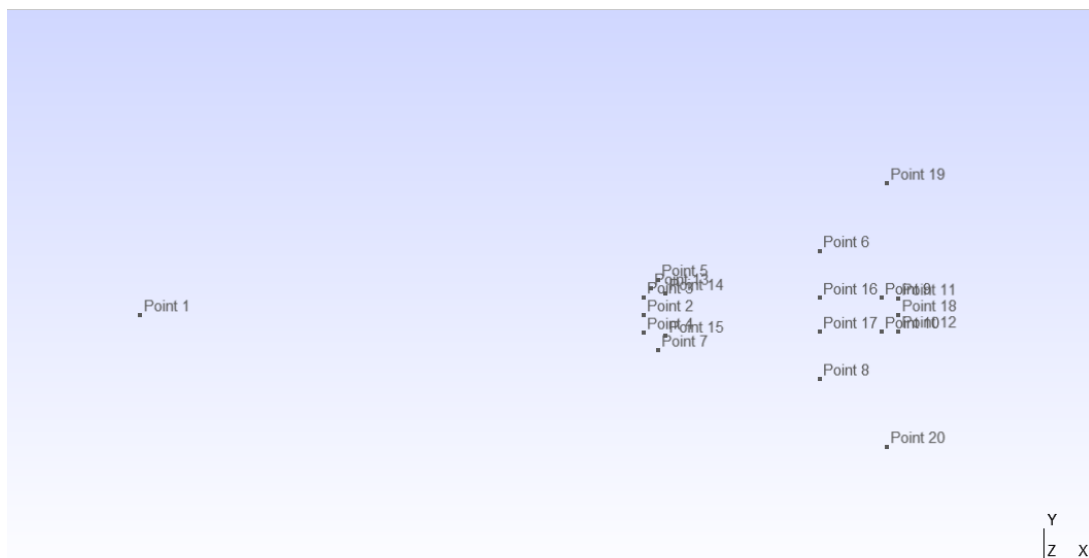
```

14 Point(9) = {43.74, 1, 0, mesh_normal};
15 Point(10) = {43.74, -1, 0, mesh_normal};
16 Point(11) = {44.75*Cos(1.23*Pi/180), 44.75*Sin(1.23*Pi/180), 0, mesh_gap
    };
17 Point(12) = {44.75*Cos(-1.23*Pi/180), 44.75*Sin(-1.23*Pi/180), 0,
    mesh_gap};
18 Point(13) = {30.1299, 1.5649, 0, mesh_normal};
19 Point(14) = {31, 1.25, 0, mesh_normal};
20 Point(15) = {31.0, -1.25, 0, mesh_normal};
21 Point(16) = {40.09, 1, 0, mesh_normal};
22 Point(17) = {40.09, -1, 0, mesh_normal};
23 Point(18) = {44.75, 0, 0, mesh_gap};
24 Point(19) = {44.75*Cos(10*Pi/180), 44.75*Sin(10*Pi/180), 0, mesh_gap};
25 Point(20) = {44.75*Cos(-10*Pi/180), 44.75*Sin(-10*Pi/180), 0, mesh_gap};

```

Listing 3.2 – Geometria do Estator

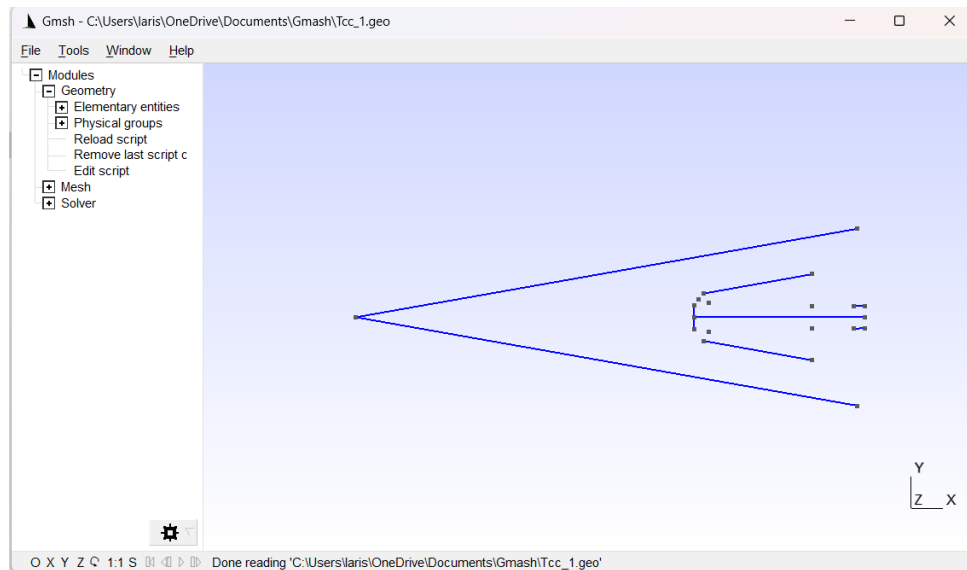
Figura 11 – Os 20 pontos definidos



Fonte: Autora, 2025.

Com os pontos previamente definidos, iniciou-se a construção da estrutura geométrica propriamente dita. Para isso, deve-se acessar o menu Geometry > Elementary entities > Add > Line, e o processo pode ser visto na figura 12.

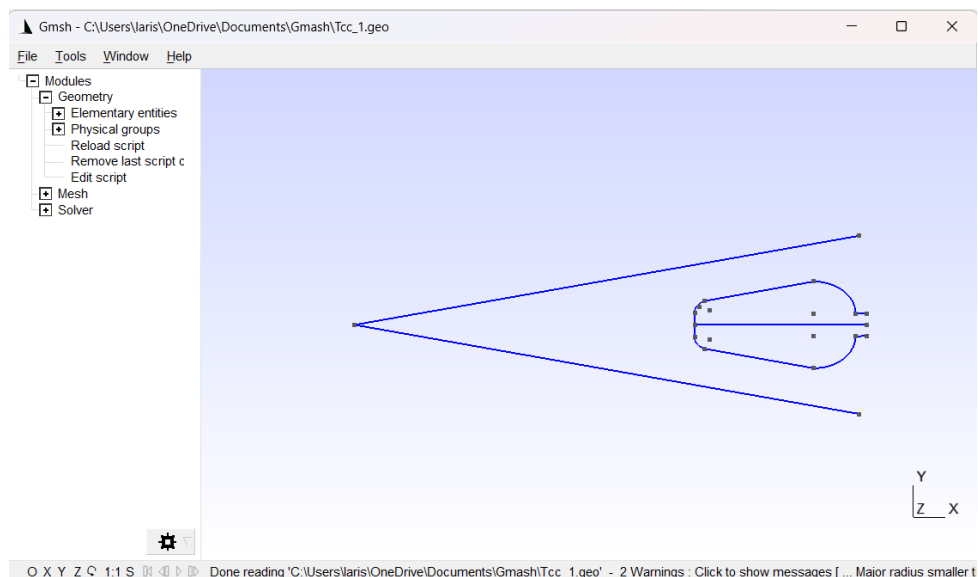
Figura 12 – Desenhando linhas e conectando os pontos



Fonte: Autora, 2025.

Ainda nessa configuração do Gmsh, seguindo o mesmo caminho, porém utilizando Geometry > Elementary entities > Add > Ellipse Arc, foram definidos os contornos dos dentes do estator, conforme ilustrado na Figura 13.

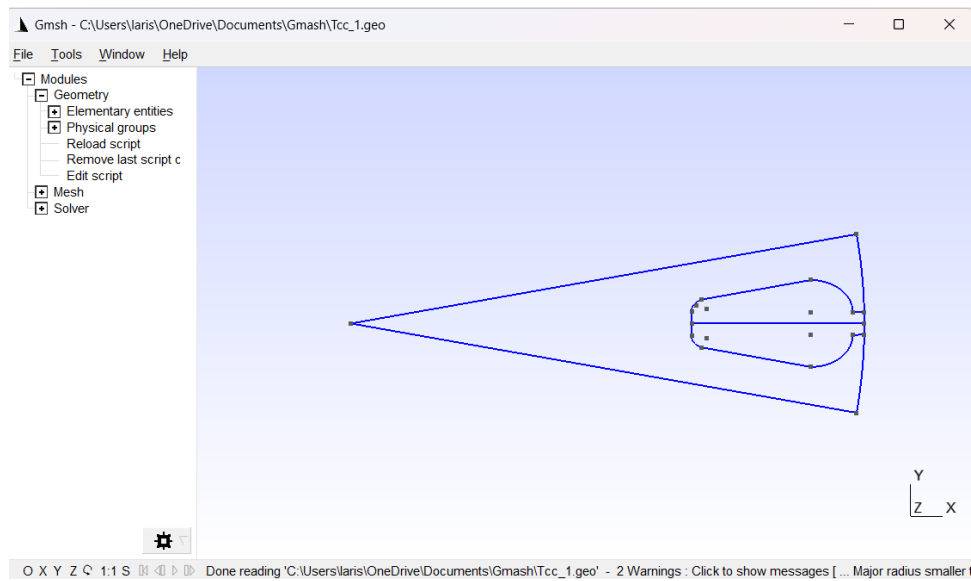
Figura 13 – Desenhando elipses



Fonte: Autora, 2025.

Em seguida, utilizando Geometry > Elementary entities > Add > Circle Arc, realizou-se o fechamento do desenho, conforme ilustrado na Figura 14.

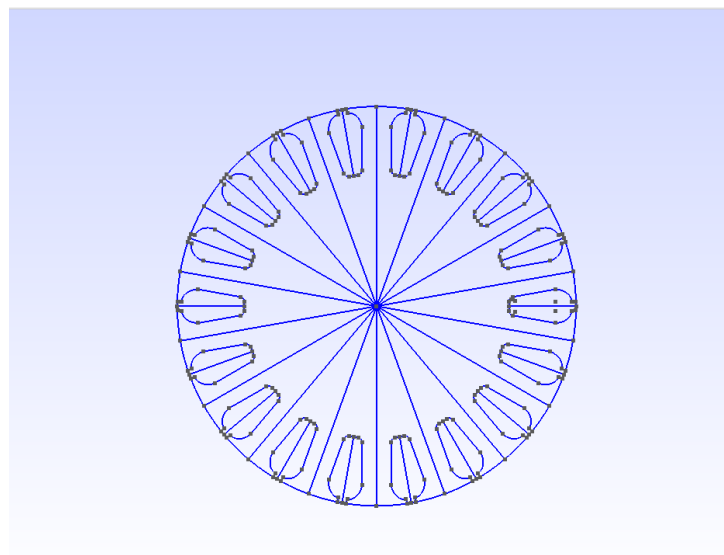
Figura 14 – Desenhando semi círculos



Fonte: Autora, 2025.

Por fim, selecciona-se a superfície utilizando o menu `Geometry > Elementary entities > Add > Plane Surface`, a fim de mapear os pontos que compõem o desenho previamente definido. Em seguida, essa superfície é inserida dentro de um laço `for`, no qual n varia de 1 a 17. Dessa forma, obtém-se a superfície completa do estator, conforme ilustrado na Figura 15, e o código correspondente pode ser visto logo abaixo.

Figura 15 – Estrutura primária



Fonte: Autora, 2025.

```
1 Line(1) = {3, 2};
2 Line(2) = {2, 4};
3 Line(3) = {5, 6};
```

```

4 Line(4) = {7, 8};
5 Line(5) = {9, 11};
6 Line(6) = {10, 12};
7 Line(7) = {2, 18};
8 Line(8) = {1, 19};
9 Line(9) = {1, 20};
10 Ellipse(50) = {3, 14, 5};
11 Ellipse(51) = {4, 15, 7};
12 Ellipse(52) = {6, 16, 9};
13 Ellipse(53) = {8, 17, 10};
14 Circle(100) = {20, 1, 12};
15 Circle(101) = {12, 1, 18};
16 Circle(102) = {18, 1, 11};
17 Circle(103) = {11, 1, 19};
18 //Definindo superficie do aco eletrico
19 Line Loop(21) = {100, -6, -53, -4, -51, -2, -1, 50, 3, 52, 5, 103, -8,
    9};
20 Plane Surface(22) = {21};
21 //Definindo superficie dos enrolamentos
22 Line Loop(23) = {102, -5, -52, -3, -50, 1, 7};
23 Plane Surface(24) = {23};
24 Line Loop(25) = {101, -7, 2, 51, 4, 53, 6};
25 Plane Surface(26) = {25};
26 //copia da estrutura primaria definindo o estator
27 For it In {1:17}
28 Rotate {{0, 0, 1}, {0, 0, 0}, it*Pi/9} {
29 Duplicata { Surface{22, 24, 26}; }
30 }
31 EndFor

```

Listing 3.3 – Desenho e configuração da geometria completa do Estator

3.1.2 Geometria do Rotor

O rotor da máquina em estudo é externo ao estator. O trecho a seguir define oito pontos que compõem parte da geometria do rotor, especificamente os limites de um segmento angular que representa um ímã permanente. Os pontos são calculados com base em coordenadas polares, utilizando os ângulos de $\pm 8,47^\circ$ e $\pm 9^\circ$ e os raios correspondentes ao limite interno do rotor ($R_{\text{rot in}}$) e à extremidade externa do ímã ($R_{\text{rot in}} + h_{\text{PM}}$). Essa definição permite gerar uma seção do rotor que foi posteriormente conectada por curvas e replicada angularmente para formar a geometria completa.

Inicialmente definimos os pontos, como pode ser verificado na Figura 16 e no código abaixo.

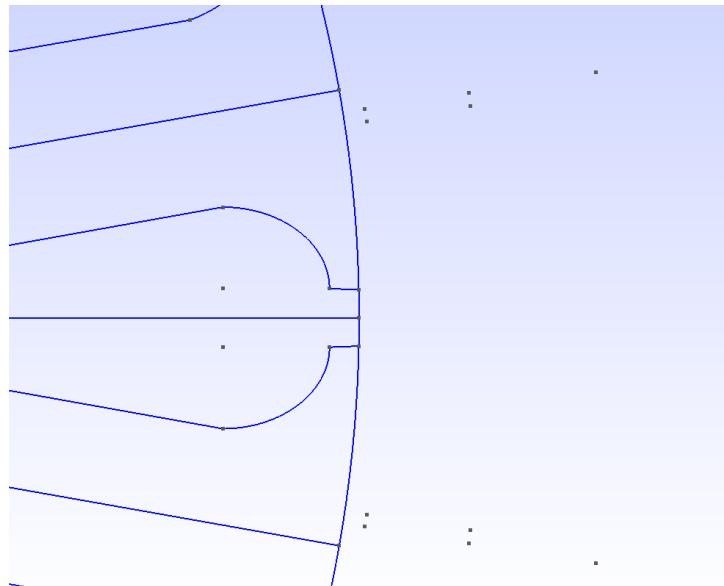
```

3 // =====
4 // ===== Geometria do Rotor =====
5 // =====
6
7 Point(153) = {R_rot_in*cos(8.47*Pi/180), R_rot_in*sin(8.47*Pi/180), 0,
    mesh_gap};
8 Point(144) = {R_rot_in*cos(-8.47*Pi/180), R_rot_in*sin(-8.47*Pi/180), 0,
    mesh_gap};
9 Point(145) = {(R_rot_in+h_PM)*cos(8.47*Pi/180), (R_rot_in+h_PM)*sin
    (8.47*Pi/180), 0, mesh_normal};
10 Point(146) = {(R_rot_in+h_PM)*cos(-8.47*Pi/180), (R_rot_in+h_PM)*sin
    (-8.47*Pi/180), 0, mesh_normal};
11
12 Point(155) = {R_rot_in*cos(9*Pi/180), R_rot_in*sin(9*Pi/180), 0,
    mesh_gap};
13 Point(147) = {R_rot_in*cos(-9*Pi/180), R_rot_in*sin(-9*Pi/180), 0,
    mesh_gap};
14 Point(148) = {(R_rot_in+h_PM)*cos(9*Pi/180), (R_rot_in+h_PM)*sin(9*Pi
    /180), 0, mesh_normal};
15 Point(149) = {(R_rot_in+h_PM)*cos(-9*Pi/180), (R_rot_in+h_PM)*sin(-9*Pi
    /180), 0, mesh_normal};

```

Listing 3.4 – Geração dos pontos iniciais para estrutura do rotor

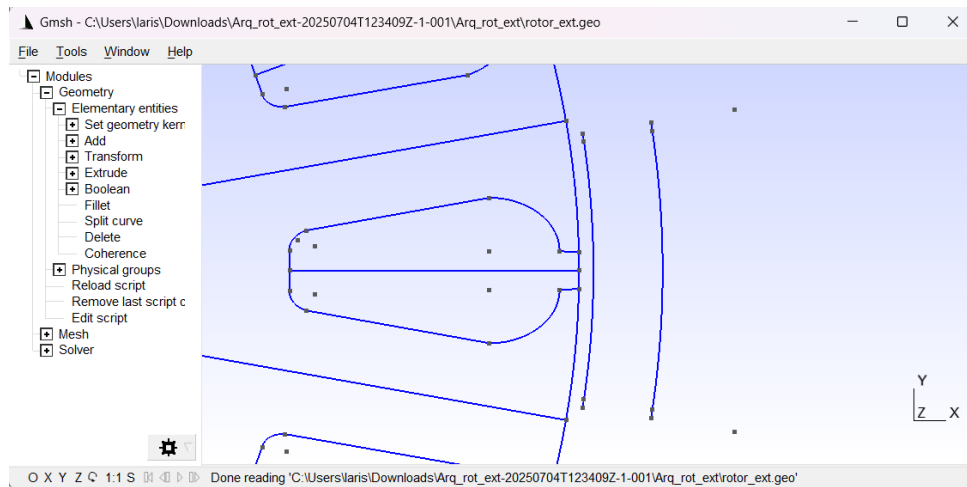
Figura 16 – Pontos iniciais para geometria do rotor



Fonte: Autora, 2025.

Em seguida, foram traçados os semicírculos conectando os pontos de ponto a ponto, de acordo com a figura 17.

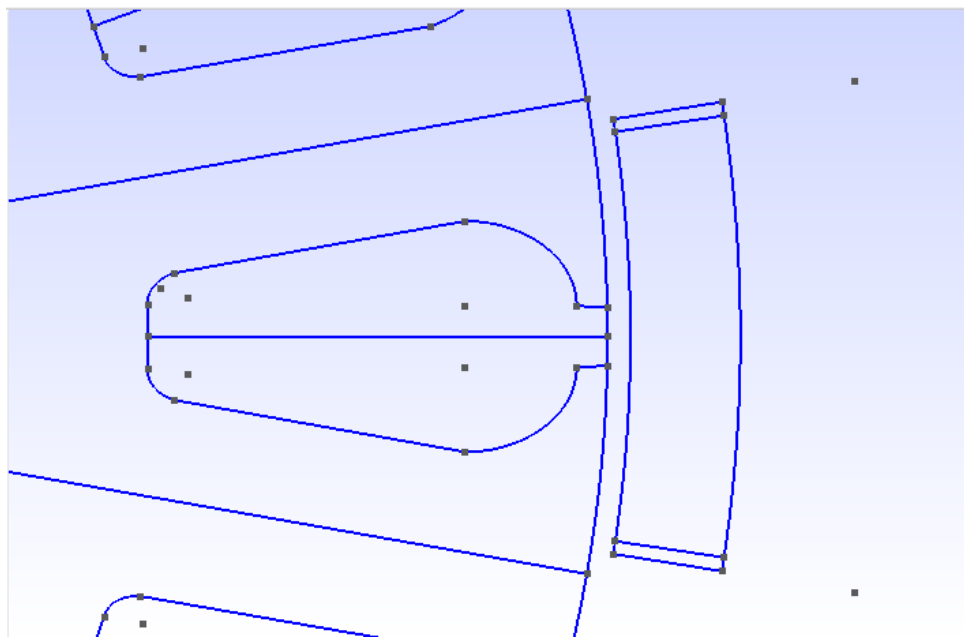
Figura 17 – Conectando os pontos com semicírculos



Fonte: Autora, 2025.

Depois, utilizam-se linhas para conectar os pontos na direção horizontal, conforme ilustrado na Figura 18.

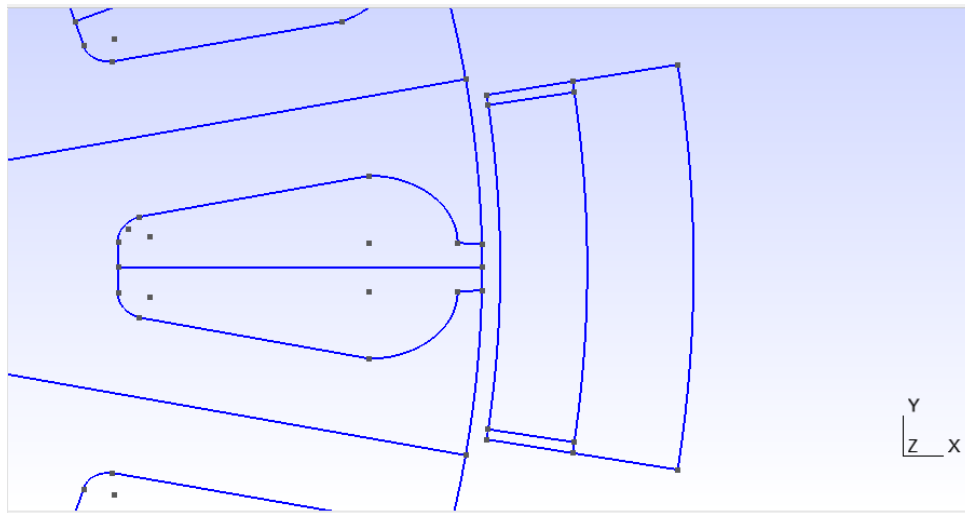
Figura 18 – Conectando os pontos com linhas



Fonte: Autora, 2025.

Após isso, utilizou-se um arco semicircular para completar essa superfície primária, de acordo com Figura 19. Vale ressaltar que essas etapas podem ser realizadas tanto via script quanto por meio do menu principal do Gmsh. No entanto, é por meio do script que se pode verificar com maior clareza se os pontos, linhas e círculos foram definidos corretamente.

Figura 19 – Estrutura primária do rotor



Fonte: Autora, 2025.

Por fim, similarmente ao procedimento feito no estator, as superfícies de interesse foram selecionadas usando o comando *plane surface* e repetidas em um laço, com isso, obteve-se a estrutura completa do rotor segundo Figura 8, já apresentada no início do capítulo e as manipulações em geral podem ser acompanhadas no código.

```

1 Circle(104) = {155, 1, 153};
2 Circle(118) = {153, 1, 144};
3 Circle(119) = {144, 1, 147};
4 Circle(121) = {145, 1, 148};
5 Circle(122) = {146, 1, 145};
6 Circle(123) = {149, 1, 146};
7 Line(10) = {147, 149};
8 Line(11) = {144, 146};
9 Line(12) = {145, 153};
10 Line(13) = {148, 155};
11 Point(150) = {(R_rot_out)*Cos(9*Pi/180), (R_rot_out)*Sin(9*Pi/180), 0,
    mesh_normal};
12 Point(151) = {(R_rot_out)*Cos(-9*Pi/180), (R_rot_out)*Sin(-9*Pi/180), 0,
    mesh_normal};
13 Line(14) = {149, 151};
14 Line(15) = {150, 148};
15 Circle(124) = {151, 1, 150};
16 //Definindo superficie do aco eletrico
17 Line Loop(26) = {14, 124, 15, -121, -122, -123};
18 Plane Surface(27) = {26};
19 //Definindo superficie do ima
20 Line Loop(28) = {122, 12, 118, 11};
21 Plane Surface(29) = {28};
22 //Definindo superficie de ar entre os imas
23 Line Loop(30) = {121, 13, 104, -12};

```

```

24 Plane Surface(31) = {30};
25 Line Loop(32) = {123, -11, 119, 10};
26 Plane Surface(33) = {32};
27 //Replica o rotor
28 For it In {1:19}
29 Rotate {{0, 0, 1}, {0, 0, 0}, it*Pi/10} {
30 Duplicata { Surface{27, 29, 31, 33}; }
31 }
32 EndFor

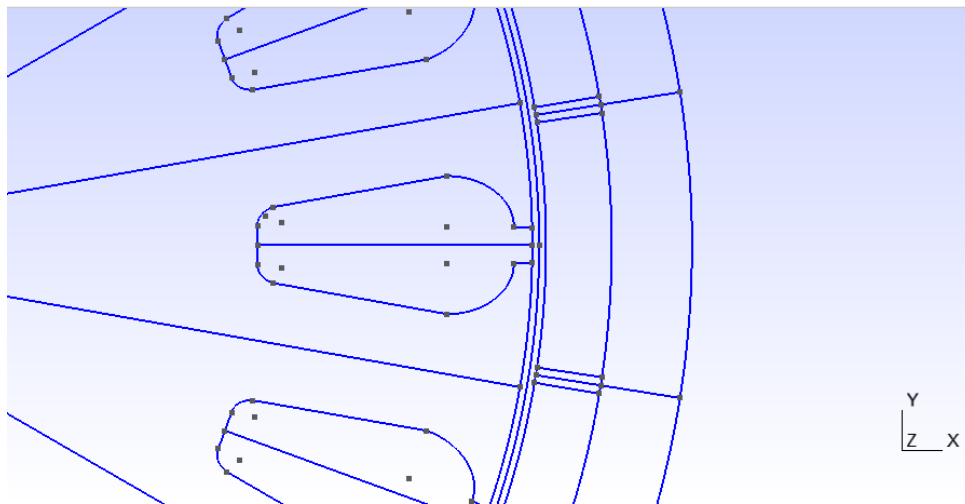
```

Listing 3.5 – Definição dos pontos para o ímã permanente

3.1.3 Entreferro

Prosseguindo com a modelagem, foi definida a geometria do entreferro da máquina, ou seja, a região de ar situada entre o estator e o rotor. Primeiramente, foram criados pontos e arcos que compõem a região central do entreferro. A seguir, definiu-se o contorno interno do entreferro do lado do estator por meio de uma sequência de curvas. A superfície do entreferro do estator foi delimitada entre esse contorno interno e a circunferência central, formando um anel em conformidade com o código e a figura 20. A mesma lógica é aplicada para a região do entreferro próxima do rotor.

Figura 20 – Entreferro



Fonte: Autora, 2025.

```

1 // =====
2 // ===== Entreferro =====
3 // =====
4 //definindo pontos e curvas
5 Point(929) = {Gap/2+R_stat_out, 0, 0, mesh_gap};
6 Point(930) = {0, Gap/2+R_stat_out, 0, mesh_gap};
7 Point(931) = {0, -Gap/2-R_stat_out, 0, mesh_gap};

```

```

8 Point(933) = {-Gap/2-R_stat_out, 0, 0, mesh_gap};
9 Circle(956) = {929, 1, 930};
10 Circle(957) = {930, 1, 933};
11 Circle(958) = {933, 1, 931};
12 Circle(959) = {931, 1, 929};
13
14 //Entreferro no estator
15 Line Loop(240) = {102, 103, 105, 128, 120, 116, 130, 153, 145, 141, 155,
    178, 170, 166, 180, 203, 195, 191, 205, 228, 220, 216, 230, 253,
    245, 241, 255, 278, 270, 266, 280, 303, 295, 291, 305, 328, 320, 316,
    330, 353, 345, 341, 355, 378, 370, 366, 380, 403, 395, 391, 405,
    428, 420, 416, 430, 453, 445, 441, 455, 478, 470, 466, 480, 503, 495,
    491, 505, 528, 520, 516, 100, 101};
16 Line Loop(241) = {956, 957, 958, 959};
17 Plane Surface(242) = {240, 241};
18 //Entreferro no rotor
19 Line Loop(243) = {118, 104, 549, 539, 544, 570, 560, 565, 591, 581, 586,
    612, 602, 607, 633, 623, 628, 654, 644, 649, 675, 665, 670, 696,
    686, 691, 717, 707, 712, 738, 728, 733, 759, 749, 754, 780, 770, 775,
    801, 791, 796, 822, 812, 817, 843, 833, 838, 864, 854, 859, 885,
    875, 880, 906, 896, 901, 927, 917, 922, 119};
20 Plane Surface(246) = {241, 243};

```

Listing 3.6 – Geometria do Entreferro

3.1.4 Definição das Regiões Físicas

A definição das regiões físicas do modelo tem como objetivo permitir a aplicação das propriedades dos materiais, densidades de corrente e condições de contorno nos cálculos. No Gmsh, essa definição consiste em associar regiões da malha às respectivas funções eletromagnéticas ou estruturais. Após a construção da geometria e a geração das superfícies por meio do comando *Plane Surface*, cada superfície é atribuída a um grupo físico com base em sua localização e função.

Além dos enrolamentos, foram definidos grupos específicos para o aço elétrico do estator, rotor, ímãs permanentes, ar entre os ímãs e o entreferro. Essa organização facilita a exportação da malha para softwares de cálculo, permitindo a associação direta das propriedades físicas.

As superfícies associadas aos enrolamentos do estator foram organizadas em três grupos distintos, correspondentes às fases da alimentação elétrica (A, B e C). Para representar o sentido da corrente elétrica, cada fase foi subdividida em dois conjuntos: positivo e negativo (por exemplo, Fase A+ e Fase A-). Esses conjuntos foram definidos utilizando o comando *Physical Surface*, aplicado a subconjuntos de superfícies previamente construídas para cada setor da máquina.

```

1 // =====

```

```

2 // ===== Superficies =====
3 // =====
4 //Fase A+-
5 Physical Surface(1) = {452, 444, 502, 494, 26, 24};
6 Physical Surface(2) = {419, 477, 469, 527, 519, 127};
7 Color Red {Surface{452, 444, 502, 494, 26, 24};}
8 Color Red {Surface{419, 477, 469, 527, 519, 127};}
9 // Fase C+-
10 Physical Surface(3) = {152, 144, 202, 194, 252, 244};
11 Physical Surface(4) = {119, 177, 169, 227, 219, 277};
12 Color Green {Surface{152, 144, 202, 194, 252, 244};}
13 Color Green {Surface{119, 177, 169, 227, 219, 277};}
14 // B+-
15 Physical Surface(5) = {302, 294, 352, 344, 402, 394};
16 Physical Surface(6) = {269, 327, 319, 377, 369, 427};
17 Color Blue {Surface{302, 294, 352, 344, 402, 394};}
18 Color Blue {Surface{269, 327, 319, 377, 369, 427};}
19 //Ferro Estator
20 Physical Surface(7) = {22, 104, 129, 154, 179, 204, 229, 254, 279, 304,
    329, 354, //379, 404, 429, 454, 479, 504};
21 Color Orange {Surface{22, 104, 129, 154, 179, 204, 229, 254, 279, 304,
    329, 354, //379, 404, 429, 454, 479, 504};}
22 //Ferro Rotor
23 Physical Surface(8) = {27, 529, 550, 571, 592, 613, 634, 655, 676, 697,
    718, 739, //760, 781, 802, 823, 844, 865, 886, 907, 27};
24 Color Orange {Surface{27, 529, 550, 571, 592, 613, 634, 655, 676, 697,
    718, 739, //760, 781, 802, 823, 844, 865, 886, 907, 27};}
25 //Ar entre os imas
26 Physical Surface(9) = {31, 546, 541, 567, 562, 588, 583, 609, 604, 630,
    625, 651, //646, 672, 667, 693, 688, 714, 709, 735, 730, 756, 751,
    777, 772, 798, 793, 819, 814, //840, 835, 861, 856, 882, 877, 903,
    898, 924, 919, 33};
27 Color Magenta {Surface{31, 546, 541, 567, 562, 588, 583, 609, 604, 630,
    625, 651, //646, 672, 667, 693, 688, 714, 709, 735, 730, 756, 751,
    777, 772, 798, 793, 819, 814, //840, 835, 861, 856, 882, 877, 903,
    898, 924, 919, 33};}
28 //Entreferro Estator
29 Physical Surface(10) = {242};
30 Color Black {Surface{242};}
31 //Entreferro Rotor
32 Physical Surface(11) = {246};
33 Color Black {Surface{246};}
34 //Imas
35 Physical Surface(12) = {29};
36 Physical Surface(13) = {536};
37 Physical Surface(14) = {557};
38 Physical Surface(15) = {578};

```

```

39 Physical Surface(16) = {599};
40 Physical Surface(17) = {620};
41 Physical Surface(18) = {641};
42 Physical Surface(19) = {662};
43 Physical Surface(20) = {683};
44 Physical Surface(21) = {704};
45 Physical Surface(22) = {725};
46 Physical Surface(23) = {746};
47 Physical Surface(24) = {767};
48 Physical Surface(25) = {788};
49 Physical Surface(26) = {809};
50 Physical Surface(27) = {830};
51 Physical Surface(28) = {851};
52 Physical Surface(29) = {872};
53 Physical Surface(30) = {893};
54 Physical Surface(31) = {914};
55 Color Cyan {Surface{29, 557, 599, 641, 683, 725, 767, 809, 851, 893};}
56 Color Yellow {Surface{536, 578, 620, 662, 704, 746, 788, 830, 872,
    914};}

```

Listing 3.7 – Definição da superfícies

3.1.5 Geração da Malha no Gmsh

O Gmsh permite a criação de malhas sofisticadas em modelos 2D e 3D. Neste projeto, o controle da densidade da malha foi realizado de forma simples, por meio da definição de pontos com diferentes níveis de densidade:

```

//limite externo
Physical Line(1) = {552, 573, 594, 615, 636, 657,
678, 699, 720, 741, 762, 783, 804, 825, 846, 867,
888, 909, 124, 531};
//limite deslizante
Physical Line(2) = {956, 957, 958, 959};
// Pontos de Densidade da malha
DefineConstant[ mesh_gap = { Gap/2}];
DefineConstant[ mesh_fine = { 0.5}];
DefineConstant[ mesh_normal = { 3}];
DefineConstant[ mesh_coarse = { 10}];

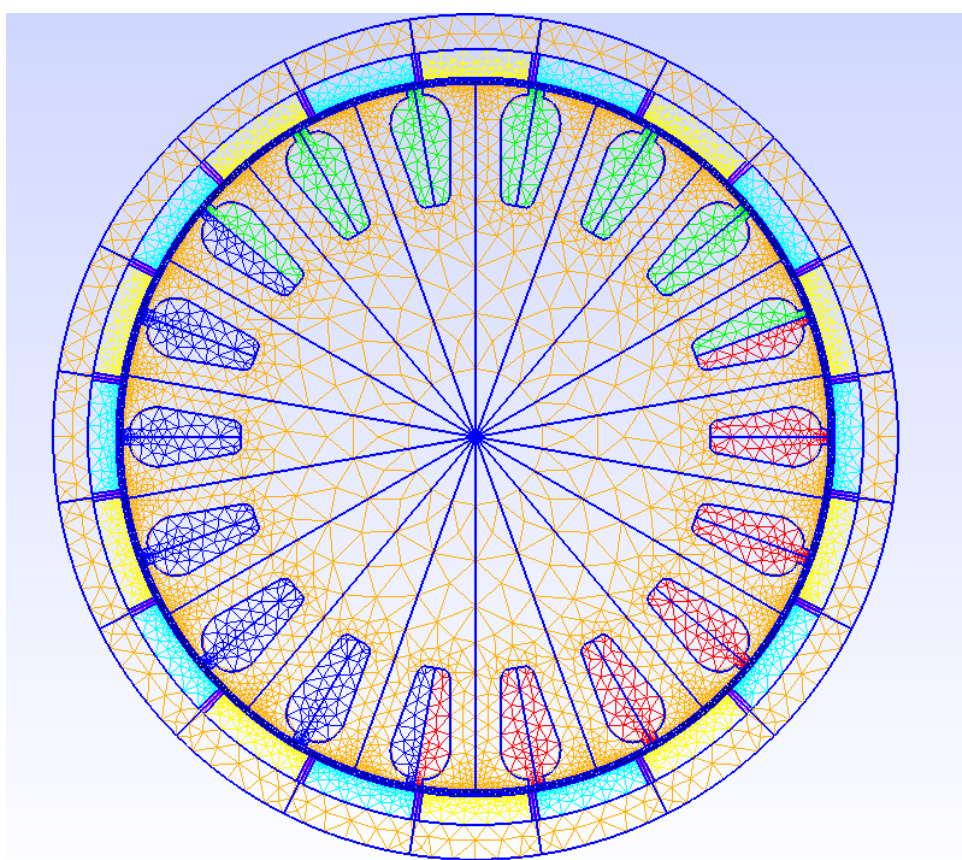
```

Para obter resultados precisos nos cálculos, a densidade da malha deve ser suficientemente alta, de modo a aproximar a solução numérica do comportamento contínuo do campo físico. No entanto, uma malha com muitos elementos aumenta significativamente o tempo computacional. Dessa forma, é necessário realizar uma

escolha criteriosa dos parâmetros de malha conforme a importância de cada região do domínio.

Na modelagem de máquinas elétricas, a região mais crítica é o entreferro, pois é onde se concentra a maior parte da energia magnética. Assim, recomenda-se que a malha nessa área seja a mais refinada. Por outro lado, nas regiões periféricas, como os contornos externos do domínio, onde as variações do campo magnético são menores, é possível utilizar uma malha primitiva. Para gerar a malha bidimensional, utilizou-se o menu *Geometry > Mesh > Define > 2D*. Na Figura 21, observa-se a malha gerada na estrutura completa.

Figura 21 – Malha pronta na estrutura



Fonte: Autora, 2025.

É recomendável verificar se as malhas foram criadas corretamente em toda geometria, especialmente na região do entreferro. Para isso, amplia-se essa área e inspeciona-se visualmente. Essa etapa permite confirmar se as superfícies foram definidas corretamente e se não há falhas ou discontinuidades que possam comprometer a qualidade dos cálculos.

3.2 SOFTWARE ELMER

Elmer é um software de simulação multifísica de código aberto desenvolvido principalmente pelo CSC - IT Center for Science (CSC). A ferramenta foi escolhida por ser gratuito e de fácil acesso (Ruololainen, *et al.*, 2023).

Para a execução dos cálculos no software Elmer, foi necessário a elaboração de um arquivo de entrada no formato Solver Input File (SIF), específico para a configuração do problema eletromagnético. Este primeiro arquivo SIF, cujo conteúdo completo se encontra no apêndice C, foi construído com uma configuração transiente destinada ao cálculo dos campos magnéticos dinâmicos, permitindo a extração de grandezas como a tensão induzida e o torque de cogging.

Vale ressaltar que neste modelo inicial, o motor foi calculado sem alimentação, uma vez que as correntes nas bobinas do estator foram definidas como nulas. O arquivo define: os componentes necessários para o modelo, incluindo os parâmetros do motor; as propriedades dos materiais; as condições de contorno e a especificação detalhada dos solvers que executam os cálculos numéricos.

Devido à necessidade de analisar o desempenho do motor em condições de operação nominal, um segundo arquivo .sif foi elaborado (apêndice D), configurando o cálculo do motor em carga. A distinção principal em relação ao cálculo anterior reside na introdução de funções MATC (Mesh Adaptive Tools and Controls) para a modelagem da densidade de corrente aplicada às bobinas do estator. Estas funções (Jap, Jan, etc.) representam uma alimentação trifásica por meio de uma série de Fourier que inclui harmônicas (até a nona ordem), visando uma representação mais realista da forma de onda de corrente.

Essas correntes foram aplicadas aos respectivos domínios das bobinas (Corpos 1 a 6) através das Forças de Corpo (*Body Force 2 a 7*), permitindo que o modelo transite de um estudo de campos magnéticos gerados exclusivamente pelos ímãs para um cálculo com plena carga. Essa configuração viabiliza a análise do torque dinâmico e a quantificação das perdas elétricas e magnéticas sob condições operacionais.

Após a construção dos arquivos .sif necessários para os cálculos, a etapa subsequente consiste na execução do programa. Para utilizar o Elmer, será necessário executar os comandos por meio do prompt de comando do sistema operacional. Neste trabalho, será utilizado o prompt de comando do Windows. Inicialmente, é preciso realizar o download e a instalação do software a partir do site oficial do Elmer. Após a instalação, o prompt pode ser acessado pesquisando “Prompt de Comando” na barra de busca da área de trabalho.

Com o prompt aberto, deve-se acessar a pasta onde estão os arquivos da geometria gerados no Gmsh. Isso pode ser feito utilizando o comando `cd`, seguido do caminho completo até a pasta desejada. Por exemplo: `cd C:\Users\documentos\Arq_rot_ext\Arq_rot_ext`. Após acessar o diretório correto, o próximo passo é exe-

cutar o solver do Elmer com o comando: `ElmerSolver nome.sif`, onde `nome.sif` deve ser substituído pelo arquivo de configuração do caso a ser simulado. O cálculo pode ser verificado na Figura 22, com o exemplo no prompt de comando.

Figura 22 – Cálculo pronto feito pelo Elmer, exemplo no prompt

```
Prompt de Comando
MAIN: -----
MAIN: Time: 180/181: 3.000E-02
MAIN: Estimated time left: 1.1 seconds
MAIN: -----
MAIN:
ComputeChange: NS (ITER=1) (NRM,RELC): ( 31830.465 0.68911707E-02 ) :: mgdyn2d
ComputeChange: NS (ITER=2) (NRM,RELC): ( 31831.199 0.23069253E-04 ) :: mgdyn2d
ComputeChange: NS (ITER=3) (NRM,RELC): ( 31831.303 0.32558419E-05 ) :: mgdyn2d
ComputeChange: NS (ITER=4) (NRM,RELC): ( 31831.304 0.36606910E-07 ) :: mgdyn2d
ComputeChange: NS (ITER=1) (NRM,RELC): ( 0.57405683 2.0000000 ) :: calcfields
ComputeChange: NS (ITER=2) (NRM,RELC): ( 0.78763639 0.31369702 ) :: calcfields
ComputeChange: SS (ITER=1) (NRM,RELC): ( 0.0000000 0.0000000 ) :: SaveScalars
MAIN:
MAIN: -----
MAIN: Time: 181/181: 3.017E-02
MAIN: -----
MAIN:
ComputeChange: NS (ITER=1) (NRM,RELC): ( 31581.615 0.78750373E-02 ) :: mgdyn2d
ComputeChange: NS (ITER=2) (NRM,RELC): ( 31582.196 0.18407425E-04 ) :: mgdyn2d
ComputeChange: NS (ITER=3) (NRM,RELC): ( 31582.302 0.33427319E-05 ) :: mgdyn2d
ComputeChange: NS (ITER=4) (NRM,RELC): ( 31582.303 0.28363048E-07 ) :: mgdyn2d
ComputeChange: NS (ITER=1) (NRM,RELC): ( 0.57375773 2.0000000 ) :: calcfields
ComputeChange: NS (ITER=2) (NRM,RELC): ( 0.78781906 0.31443153 ) :: calcfields
ComputeChange: SS (ITER=1) (NRM,RELC): ( 0.0000000 0.0000000 ) :: SaveScalars
MAIN: *** Elmer Solver: ALL DONE ***
MAIN: The end
SOLVER TOTAL TIME(CPU,REAL): 97.43 97.43
ELMER SOLVER FINISHED AT: 2025/08/24 15:56:42
```

Fonte: Autora, 2025.

Ao final do cálculo, na pasta de destino será gerado uma pasta com o nome `results`, que será usada para ser aberta no ParaView, onde será feito os outros passos. Na figura 23, está a pasta com o nome `Results` para título de verificação.

Figura 23 – Print da pasta `results` que aparece na pasta de destino

Nome	Data de modificação	Tipo	Tamanho
results	19/08/2025 16:16	Pasta de arquivos	
rotor_ext	19/08/2025 16:03	Pasta de arquivos	
case_emf_rotext	19/08/2025 16:03	Arquivo SIF	14 KB
case_torquecorreto	19/08/2025 16:03	Arquivo SIF	19 KB
rotor_ext	19/08/2025 16:03	Arquivo GEO	10 KB
rotor_ext.msh	19/08/2025 16:03	Arquivo MSH	794 KB

Fonte: Autora, 2025.

3.3 SOFTWARE PARAVIEW

O ParaView é um software de visualização de dados científicos que permite analisar e explorar grandes volumes de dados, com foco em simulações e modelagens científicas. Ele é amplamente utilizado em áreas como engenharia, física e ciências da

computação. O ParaView oferece uma interface gráfica interativa, mas também pode ser controlado via scripts, geralmente utilizando Python, para processar e visualizar dados em 3D (PARAVIEW, 2025).

Ele suporta diversos formatos de dados e permite a realização de análises complexas, incluindo processamento de volumes, representações gráficas avançadas (como gráficos de superfícies, contornos e isosuperfícies), animações e a integração com clusters de computação para lidar com grandes conjuntos de dados. ParaView é open-source, o que o torna altamente customizável e acessível.

Em um primeiro momento, realizaram-se as configurações iniciais e os cálculos do motor sem alimentação. O objetivo dessa etapa foi validar o modelo geométrico desenvolvido e realizar o cálculo da tensão induzida por fase. Na sequência, foram analisadas as linhas de fluxo, a distribuição da indução magnética no entreferro, o torque de borda e as tensões induzidas.

3.3.1 Linhas de fluxo

O primeiro passo para iniciar a análise no ParaView consistiu em abrir o software, acessar o menu File e selecionar a opção Open. Em seguida, deve-se navegar até a pasta de destino, localizada no diretório results, e selecionar o arquivo step.vtu. A Figura 24 abaixo apresenta uma captura de tela da pasta de destino, de onde o arquivo foi extraído.

Figura 24 – Print da pasta results que aparece na pasta de destino

Nome	Data de modificação	Tipo	Tamanho
results	19/08/2025 16:16	Pasta de arquivos	
rotor_ext	19/08/2025 16:03	Pasta de arquivos	
case_emf_rotext	19/08/2025 16:03	Arquivo SIF	14 KB
case_torquecorreto	19/08/2025 16:03	Arquivo SIF	19 KB
rotor_ext	19/08/2025 16:03	Arquivo GEO	10 KB
rotor_ext.msh	19/08/2025 16:03	Arquivo MSH	794 KB

Fonte: Autora, 2025.

No ParaView, foi aberto o motor com cor sólida, como ilustrado pela figura 25. Então, na seção Properties, na aba Coloring, deve-se selecionar a opção Magnetic Flux Density. Dessa forma, o motor foi visualizado com a densidade de fluxo magnético, diferenciada por várias cores, conforme ilustrado na figura 26. A mudança no fluxo magnético pode ser observada ao pressionar o botão Play localizado no menu superior do software.

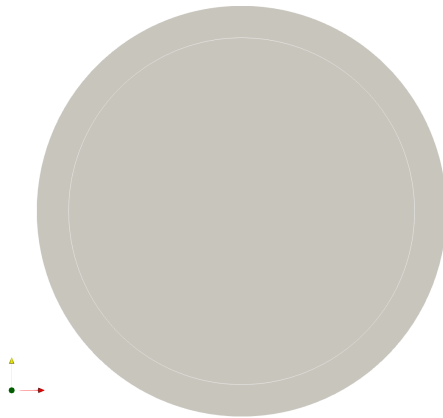


Figura 25 – Motor inicialmente com cor sólida.
Fonte: Autora, 2025.

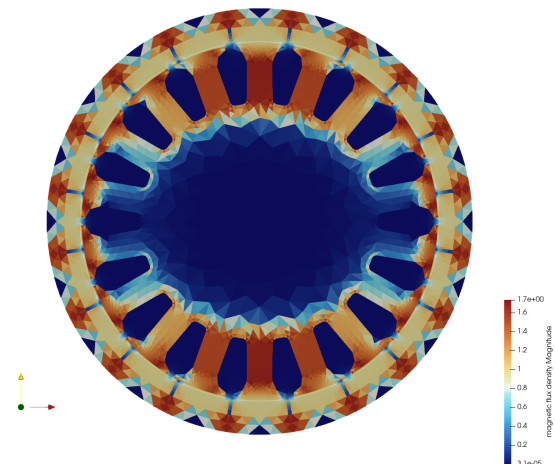
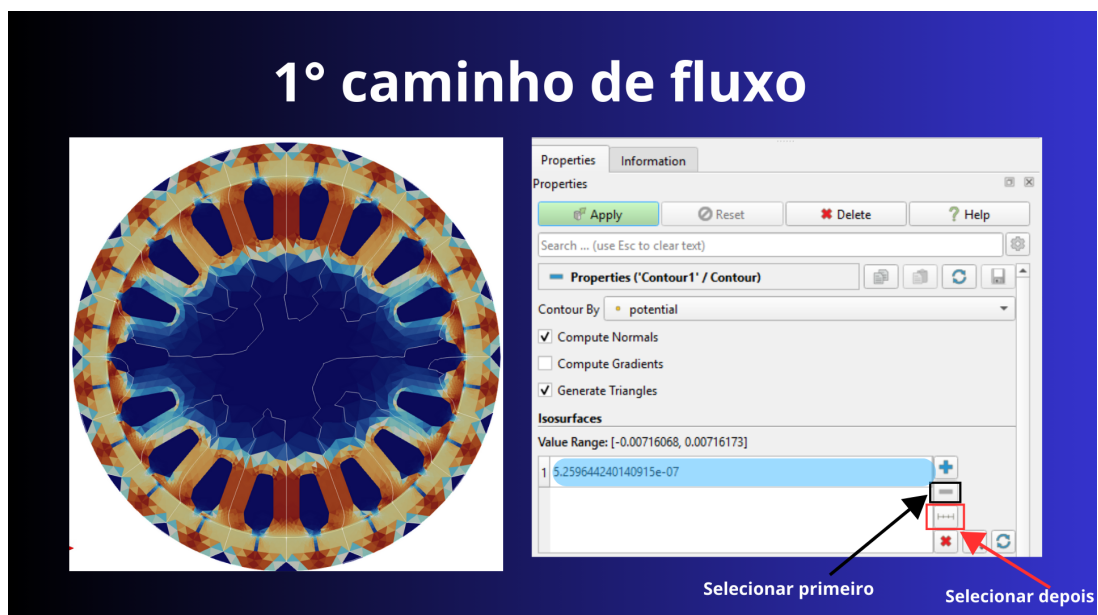


Figura 26 – Motor com mostrando a densidade de fluxo magnético.
Fonte: Autora, 2025.

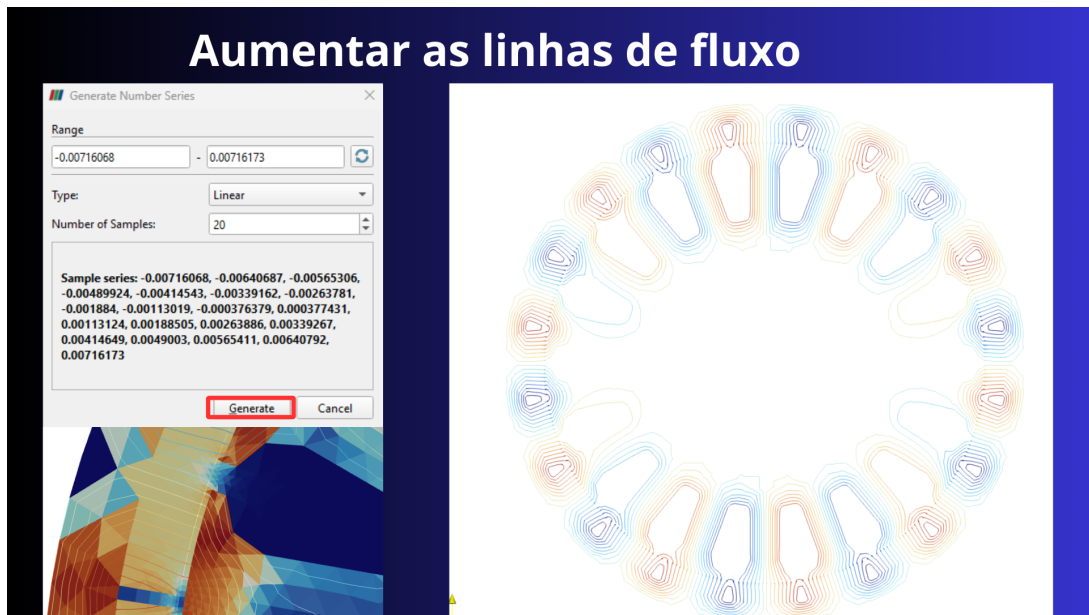
Para visualizar as linhas de fluxo, deve-se selecionar a opção `Filters > Alpha-betical > Contour`. Esse comando criou um único caminho, conforme ilustrado na Figura 27. No entanto, é possível modificar essa configuração acessando a seção `Properties (Contour) > Isosurfaces`. No intervalo de valores (`Value Range`), foi exibido um único valor. Deve-se selecioná-lo e, em seguida, clicar no botão com o símbolo de menos (-). Após essa alteração, deve-se clicar no ícone de régua, o que abriu uma nova janela, na qual, é necessário ajustar o campo `Number of samples` de 10 para 20. Por fim, deve-se selecionar a opção `Generate`, conforme o passo a passo ilustrado na Figura 28.

Figura 27 – Primeiro caminho de fluxo e configuração para criar-se mais caminhos



Fonte: Autora, 2025.

Figura 28 – Resultado das configurações anteriores

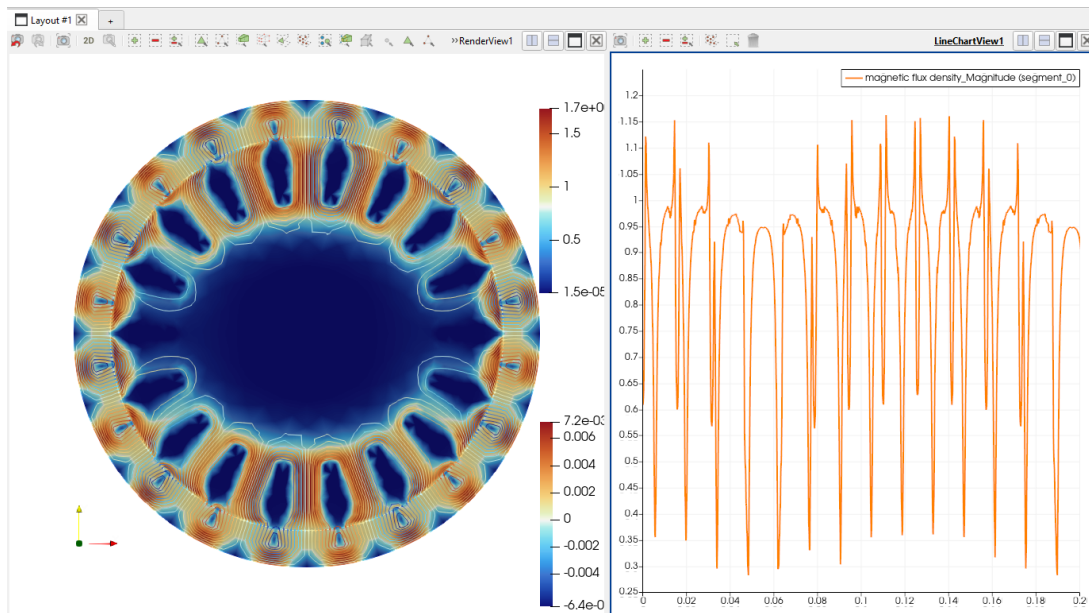


Fonte: Autora, 2025.

3.3.2 Indução magnética no entreferro

A partir da seção anterior, foi adicionado mais um filtro, seguindo o caminho `Filters > Alphabetical > Contour > CellDatatoPointData`. Em seguida, deve-se selecionar a ferramenta `CellDatatoPointData` e aplicar sobre ele um novo recurso, acessível pelo caminho `Filters > Alphabetical > Plot on Intersection Curves`. É necessário, ainda, modificar os seguintes campos: `Slice type`: sphere; `Center`: centro da esfera; `Radius`: raio no meio do entreferro. Essa operação criou um gráfico contendo todas as variáveis presentes no cálculo. Como o objeto de interesse, neste caso, é a indução magnética, foi necessário desmarcar as demais variáveis no campo `Variable`, mantendo selecionada apenas a opção `Magnetic Flux Density Magnitude`, conforme Figura 29.

Figura 29 – Indução magnética no entreferro



Fonte: Autora, 2025.

A figura 29 permite observar o comportamento inicial do campo magnético no motor, destacando-se a concentração de fluxo nas extremidades dos dentes do estator. Essa característica ocorreu devido à menor relutância oferecida pelo material ferromagnético nessas regiões, em contraste com as ranhuras e o entreferro, que dificultam a condução do fluxo. É evidenciado que, mesmo em repouso, o campo tende a se organizar segundo a geometria da máquina, validando a coerência do modelo numérico.

O gráfico de densidade de fluxo confirma essa distribuição, apresentando oscilações periódicas associadas à alternância entre dentes e ranhuras. Os picos correspondem às áreas de maior concentração de fluxo, enquanto as quedas indicam as regiões de baixa intensidade magnética. Esse comportamento é esperado e serve como referência para comparações posteriores quando o motor for energizado.

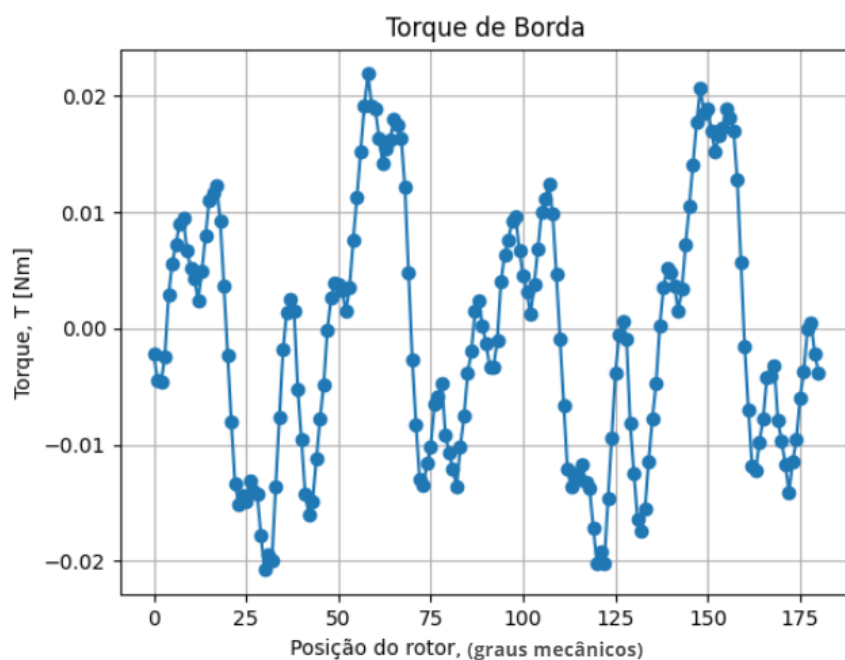
3.3.3 Torque de Borda

O software ElmerSolver salva os valores de torque de borda em arquivos de texto, que se encontra na pasta `results` e se chama `scalars.dat`, que registram os resultados escalares organizados por passo de tempo. Esses arquivos, embora contenham as informações necessárias, não são práticos para visualização direta no ParaView, pois este é dedicado principalmente à visualização em malhas. Dessa forma, torna-se necessário utilizar ferramentas adicionais para processar esses dados e permitir uma análise mais clara e eficiente.

Para o pós-processamento dos dados, utilizou-se a linguagem de programação Python. As bibliotecas NumPy e SciPy foram empregadas para a execução de

cálculos matemáticos, enquanto a Matplotlib permitiu a geração de gráficos. O fluxo de trabalho consiste em localizar o arquivo de resultados (`C:/documentos/user/pasta/results/scalars.dat`), seguida pelo carregamento dos dados no ambiente Python. Posteriormente, o torque de borda ao longo do tempo foi gerado com o uso da Matplotlib. A identificação da coluna que contém os dados de interesse foi realizada por meio da consulta ao arquivo `scalars.dat.names`, que apresenta o cabeçalho correspondente. A Figura 30 ilustra o gráfico resultante e o script utilizado para sua geração está no apêndice A desde trabalho.

Figura 30 – Gráfico do torque de Borda

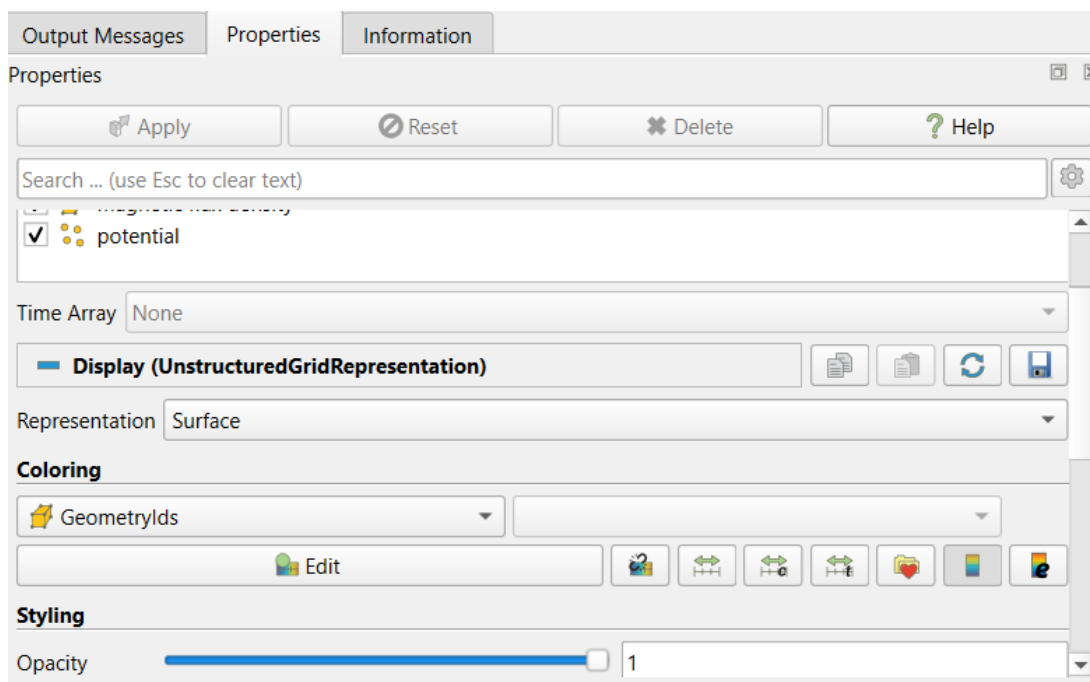


Fonte: Autora, 2025.

3.3.4 Tensão Induzida

O cálculo da tensão induzida no motor foi produzido utilizando o software Paraview. Para iniciar a análise dos dados do volume, primeiramente fez-se a abertura do arquivo `.vtu`, o qual se encontra localizado na pasta `results` do projeto do cálculo. Após a abertura bem-sucedida do mesmo, torna-se necessário preparar a visualização da geometria para o cálculo subsequente. Este passo é executado navegando-se até a opção de `Coloring` na interface do software, onde a variável de visualização padrão deve ser alterada para `GeometryIds`, de acordo com Figura 31.

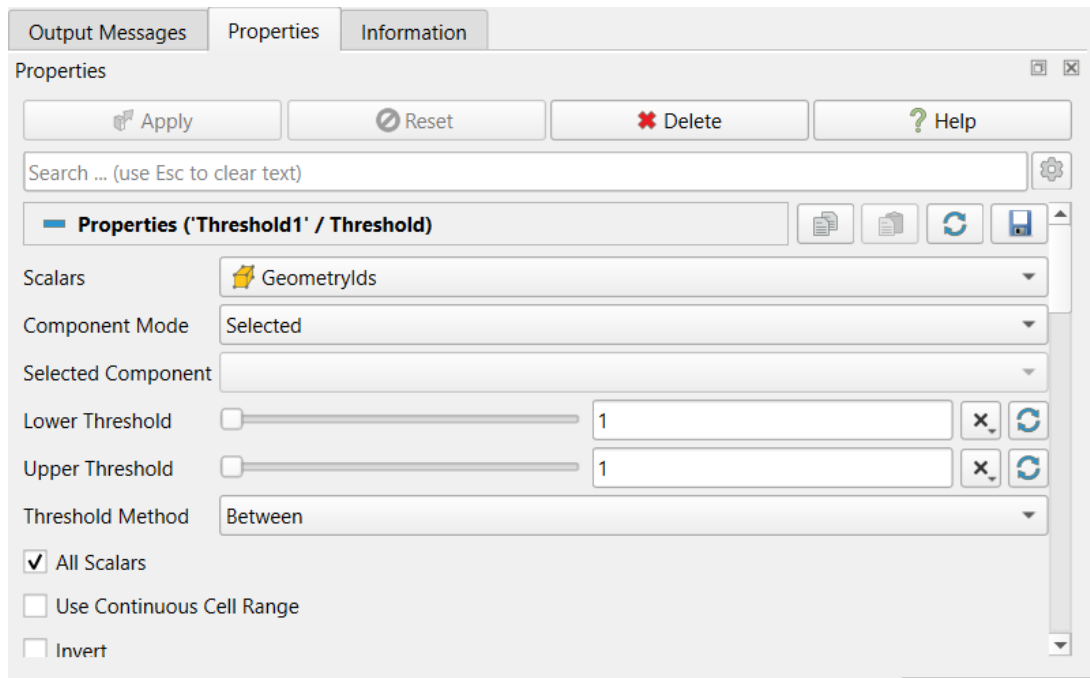
Figura 31 – Configuração - Passo 1



Fonte: Autora, 2025.

Para filtrar as regiões de interesse, como as fases $U+$ (positiva) e $U-$ (negativa), aplica-se o filtro Threshold, acessado através do caminho: Filter > Alphabetical > Threshold. Com a aplicação do filtro, uma nova aba será exibida, na qual o campo Scalars deve ser alterado para GeometryIds. Inicialmente, para o cálculo da fase positiva ($U+$), os campos Lower e Upper devem ser preenchidos com o valor 1 (numeração referente à configuração das regiões físicas no arquivo de extensão case_emf_rotoext.sif) e, em seguida, a operação deve ser confirmada, conforme a Figura 32.

Figura 32 – Configuração da Fase positiva - Passo 2

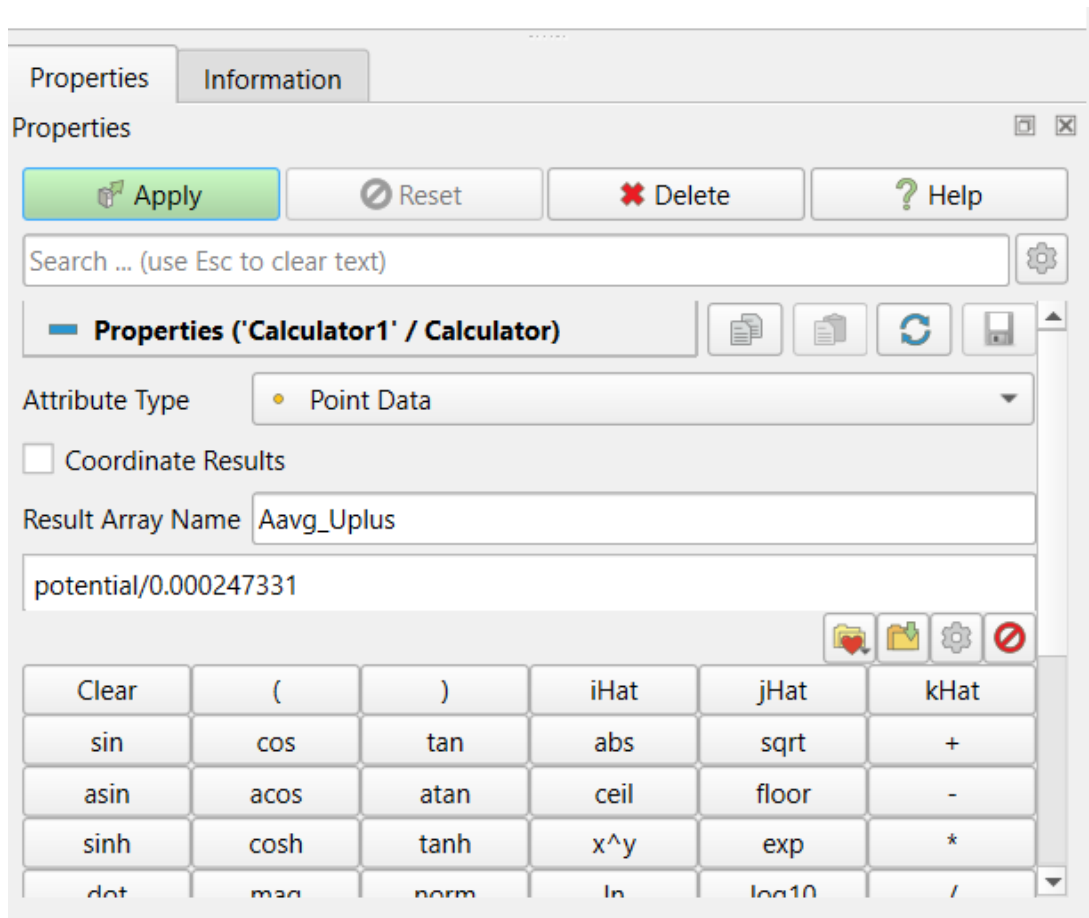


Fonte: Autora, 2025.

Com o filtro **Threshold** previamente selecionado, adiciona-se o filtro subsequente, **Integrate Variables**, localizado no menu **Filter > Alphabetical**. Na sequência, aplica-se o filtro **Calculator**, o que resultará na abertura de um novo painel de configuração.

Neste painel, uma nova variável deve ser criada com o nome **Aavg_Uplus**, e sua expressão de cálculo deve ser definida como **potential/area**, conforme figura 33. Ressalta-se que o valor da área, requerido pela expressão, é um resultado da etapa de integração e pode ser consultado na tabela de dados exibida na seção **Cell Data**.

Figura 33 – Configuração da Fase positiva - Passo 3

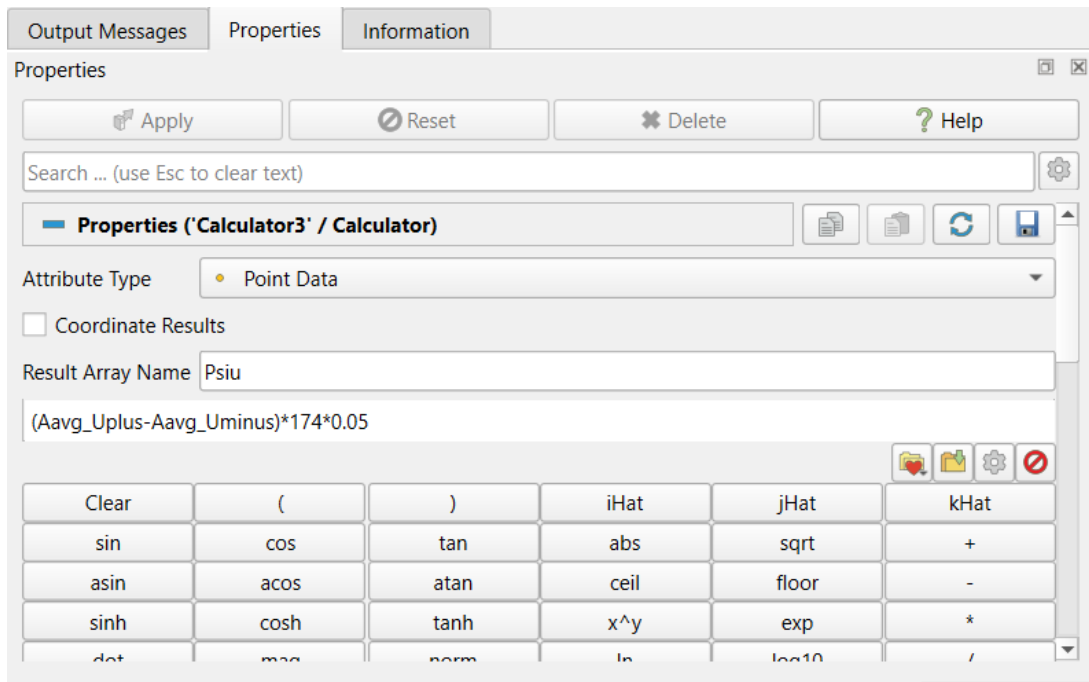


Fonte: Autora, 2025.

De forma análoga, o procedimento é replicado para a fase negativa (U-), partindo-se da configuração principal. Neste caso, ressalta-se a necessidade de inserir o valor da região física, sendo 2 neste trabalho, nos campos Lower e Upper. A variável resultante desta etapa é denominada Aavg_Uminus, calculada por meio da mesma expressão, potential/area .

Posteriormente, realiza-se a unificação dos atributos de ambos os cálculos com o filtro Append Attributes. Com as duas saídas selecionadas, aplica-se o filtro Append Attributes, localizado no caminho Filters > Alphabetical. Por fim, um filtro Calculator adicional é empregado sobre o resultado para o cálculo do fluxo concatenado. A variável de saída é nomeada Psiu, e a expressão utilizada é a multiplicação do fluxo concatenado da fase, do número de espiras por fase e da altura de pacote $(\text{Aavg_Uplus} - \text{Aavg_Uminus}) * 174 * 0.05$, conforme apresentado na Figura 34.

Figura 34 – Cálculo da Psiu



Fonte: Autora, 2025.

Feito isso, o cálculo é concluído e para mostrar a curva do fluxo seleciona-se o resultado e, por meio do caminho *Filters > Alphabetical > PlotDataOverTime*, aplica-se a última ferramenta possibilitando a visualização gráfica da curva do fluxo concatenado.

A etapa seguinte consiste na exportação dos dados para um arquivo de planilha, permitindo análises subsequentes, como o cálculo da tensão induzida em ferramentas como o Python. O procedimento de salvamento é executado pelo menu *File > Save Data*, onde o usuário deve especificar o nome do arquivo .csv. Uma ilustração deste processo é apresentada na Figura 35, que culmina na criação de uma planilha na pasta de destino.

Figura 35 – Configuração para salvar o arquivo em uma tabela

Configure Writer (CSV Writer)

☒ Write Time Steps

☐ Write Time Steps Separately

File name suffix:

☐ Choose Arrays To Write

CSV Writer Parameters

Precision:

☐ Use Scientific Notation

Field Association:

☒ Add Meta Data

☐ Add Time Step

☐ Add Time

☒ Use String Delimiter

Fonte: Autora, 2025.

O Apêndice B apresenta o código-fonte em Python utilizado no cálculo da tensão induzida e a geração dos gráficos do fluxo concatenado e de suas componentes harmônicas. Além disso, ele também salva as informações de fluxo e tensão induzida.

Ressalta-se a importância de verificar a coluna correspondente aos dados da variável Ψ_{iu} . No escopo deste trabalho, tais dados localizam-se na quinta coluna, sendo acessados pelo índice 4, em conformidade com a indexação iniciada em zero. Dando prosseguimento ao processo, o cálculo e a geração da forma de onda da tensão induzida é feita.

3.3.4.1 Motor alimentado

A excitação elétrica do motor no ambiente de cálculo foi efetuada pela imposição de uma densidade de corrente nos enrolamentos do estator. Para tal, definiram-se previamente os parâmetros fundamentais no arquivo de configuração, como a corrente de pico, o número de espiras por fase e a área da seção transversal do condutor. Tais valores foram utilizados para o cálculo da magnitude da densidade de corrente, aplicada dinamicamente ao longo dos cálculos.

Para cada uma das três fases do estator, desenvolveram-se funções matemáticas que descrevem a densidade de corrente instantânea em função do tempo, representando o formato trapezoidal e modeladas por meio da série de Fourier. Criaram-se

funções distintas para os lados positivo e negativo de cada fase, nas quais as funções para os lados negativos apresentam sinal invertido, a fim de representar corretamente o fluxo de corrente em sentido oposto.

Concluídas as etapas de pré, processamento e pós-processamento, os resultados obtidos são apresentados no capítulo subsequente.

4 RESULTADOS

Este capítulo apresenta e discute os resultados obtidos no cálculo do motor elétrico. A análise é dividida em duas seções principais: análise da tensão induzida, e a operação com o motor alimentado, na qual se investigam o comportamento do torque eletromagnético e a distribuição da densidade de fluxo magnético.

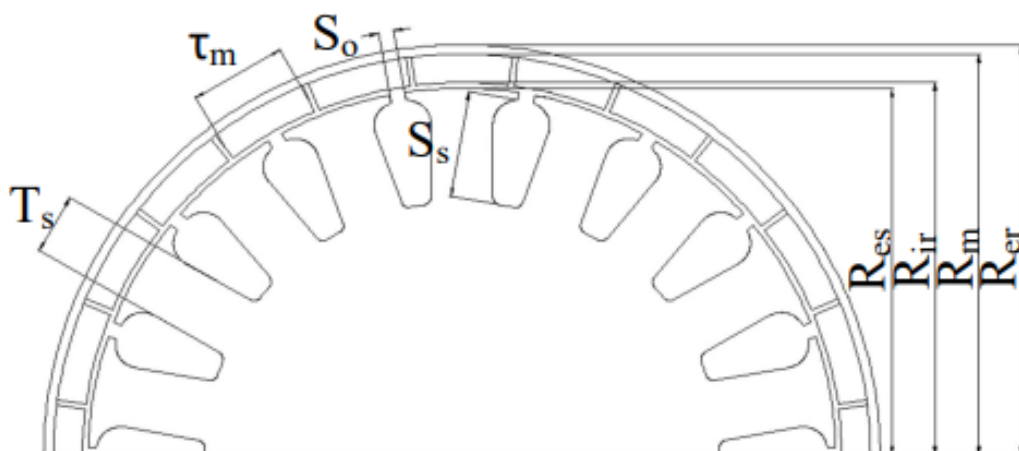
Com o objetivo de validar a metodologia numérica empregada, os resultados gerados pelo software Elmer são comparados com cálculos de referência, executados no software FEMM, uma ferramenta consolidada e amplamente utilizada para a análise de máquinas elétricas por elementos finitos desde 2014 (Meeker, 2018).

Para contextualizar a análise, a seção inicial deste capítulo detalha os parâmetros dimensionais e as principais características nominais da máquina em estudo.

4.1 DIMENSÕES DA MÁQUINA

O motor elétrico que serve como objeto de estudo e referência neste trabalho é baseado no protótipo utilizado em um pequeno veículo elétrico acadêmico e foi abordado anteriormente no trabalho de Juliani et al. (2025) . Trata-se de um motor síncrono de ímãs permanentes (MSIP) com topologia de rotor externo, uma configuração particularmente atrativa para aplicações de tração elétrica, pois permite o acoplamento direto do motor à roda do veículo. As principais características dimensionais e nominais da máquina, que servem como base para a análise e os cálculos subsequentes, estão detalhadas na Tabela 1 e 2, e na Figura 36 a seguir.

Figura 36 – Dimensões do motor com Rotor externo



Fonte: Juliani et al. (2024, p 3).

Tabela 1 – Características dimensionais do motor de rotor externo

Parâmetro	Símbolo	Rotor Externo
Altura do pacote de chapas (mm)	L_s	36
Número de polos	P	20
Número de ranhuras	N_S	18
Raio externo do estator (mm)	R_{es}	44.75
Raio interno do rotor (mm)	R_{ir}	45.50
Raio do ímã (mm)	R_m	49.1
Raio externo do rotor (mm)	R_{er}	53.5
Largura do dente do estator (mm)	T_s	6.5
Largura da abertura da ranhura (mm)	S_o	2.0
Largura do ímã (mm)	τ_m	13.45
Altura do dente do estator (mm)	S_s	14.0
Espiras por ranhura do estator	N	10
Diâmetro do fio (mm)	D_{wire}	1.71

Fonte: Adaptado de Juliani et al. (2024).

Tabela 2 – Especificações técnicas da máquina de estudo

Parâmetro	Valor
Tensão de entrada no inversor trifásico (V)	36
Faixa de potência de saída (W)	100–500
Velocidade máxima do veículo (Km/h)	29.1
Corrente RMS de partida por fase (A)	12.9
Corrente RMS estabilizada por fase (A)	5.2
Número de fases	3
Número de polos	20
Relação de transmissão	4:1
Velocidade nominal do motor sem caixa de engrenagens (rpm)	800

Fonte: Adaptado de Juliani et al. (2024).

4.2 VALIDAÇÃO DO MODELO: COMPARAÇÃO ENTRE ELMER E FEMM

Para validar a metodologia numérica e o modelo desenvolvido no software Elmer, os resultados obtidos foram comparados com os cálculos de referência executados no software FEMM.

Vale salientar que, ambos os softwares de cálculo operam sob licenciamento gratuito, sendo que a diferença entre eles reside em suas respectivas capacidades computacionais e restrições de domínio. O Software FEM está restrito à análise bidimensional (2D) de sistemas estáticos, enquanto o Elmer permite a execução de análises tridimensionais (3D) e a inclusão de cálculos transientes (dependência temporal),

conferindo-lhe maior robustez.

Esta seção detalha a ferramenta de referência e apresenta os resultados do Elmer em duas etapas: caracterização da tensão induzida e análise sob carga. Por fim, realiza a análise comparativa entre as duas plataformas.

4.2.1 Ferramenta de Referência: FEMM

O *Finite Element Method Magnetics* (FEMM) é um pacote de software de análise por elementos finitos gratuito, que foi desenvolvido para a resolução de problemas eletromagnéticos de baixa frequência. Sua aplicação se restringe a domínios bidimensionais, sejam eles planares ou axissimétricos. O programa tem como escopo a modelagem e solução de problemas magnetostáticos, problemas magnéticos harmônicos, problemas eletrostáticos e de fluxo de calor em regime permanente. Sua utilidade reside em fornecer uma ferramenta consolidada para projetar a geometria, definir as características dos materiais e analisar as soluções de campo resultantes (Meeker, 2020).

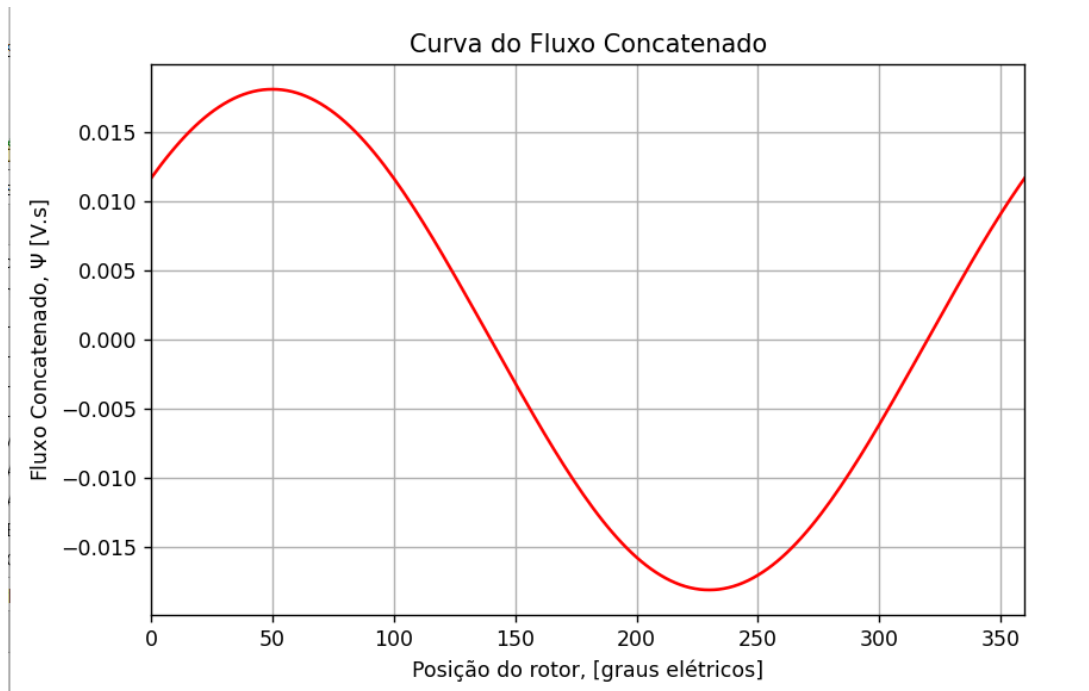
4.2.2 Análise dos cálculos no Elmer

A análise do modelo no Elmer foi dividida em duas condições de operação distintas: a caracterização das tensões induzidas pelos ímãs permanentes, e os cálculos sob carga, para avaliar o torque eletromagnético.

4.2.2.1 Caracterização das tensões induzidas (Sem Alimentação)

Nesta análise, o motor é rotacionado em sua velocidade nominal sem corrente nos enrolamentos do estator. A Figura 37 exibe a forma de onda do fluxo magnético concatenado em uma das fases, gerado exclusivamente pela rotação do rotor.

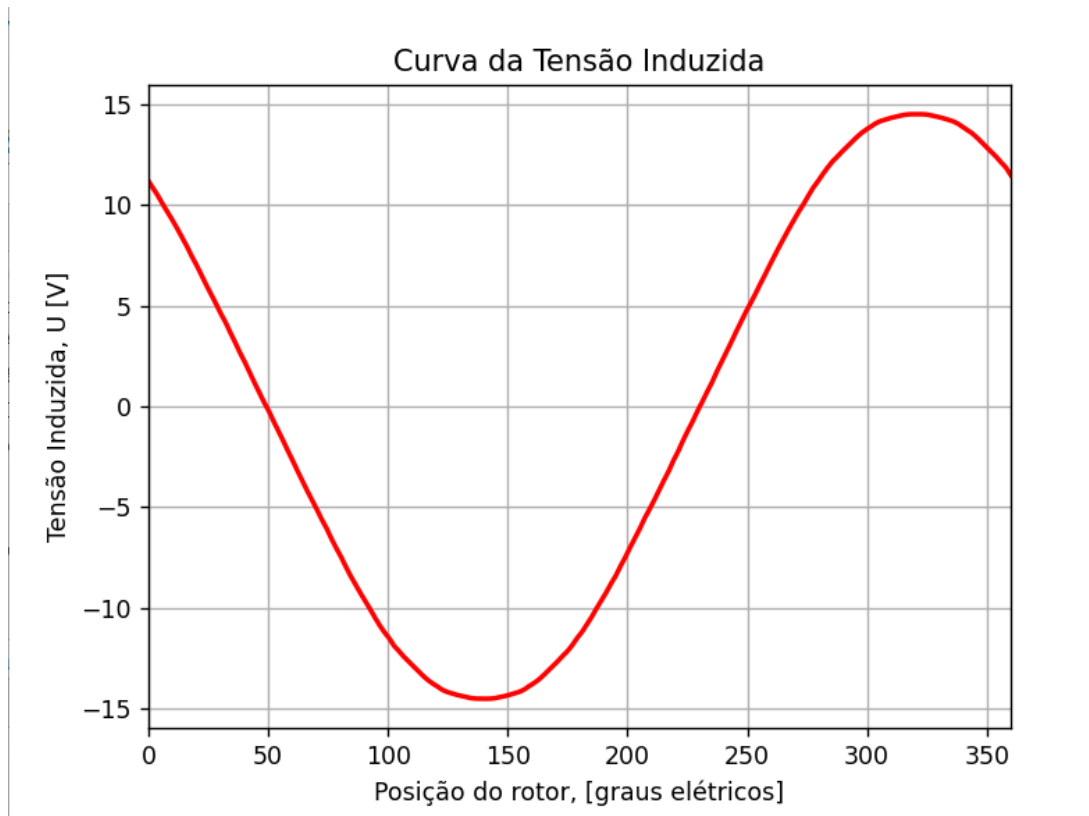
Figura 37 – Fluxo concatenado em uma fase no cálculo da tensão induzida



Fonte: Autora (2025).

A variação temporal deste fluxo induz uma força eletromotriz ou tensão induzida nos terminais da fase, cuja forma de onda é apresentada na Figura 38.

Figura 38 – Tensão induzida em uma fase (Elmer)

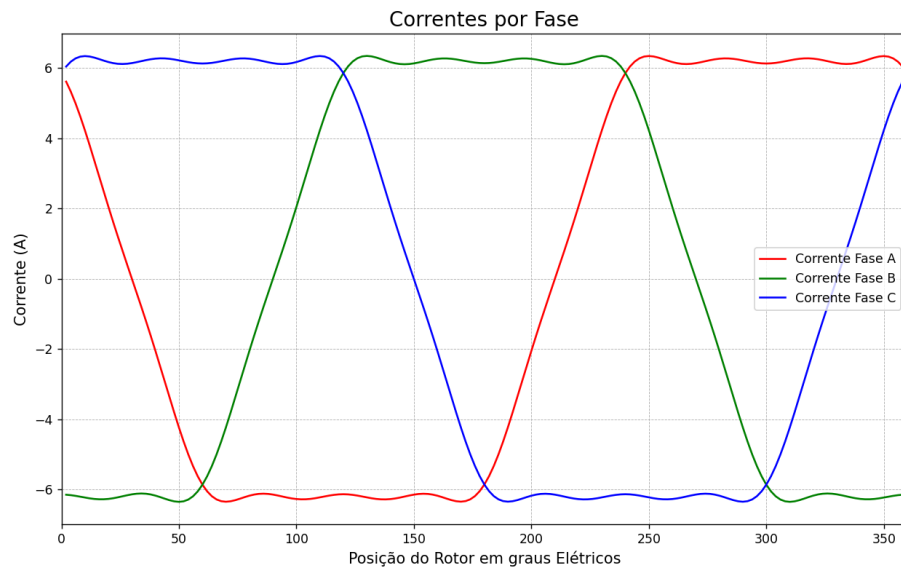


Fonte: Autora (2025).

4.2.2.2 Análise sob Carga (Com Alimentação)

Para a análise de produção de torque, o motor foi alimentado por correntes trifásicas nos enrolamentos do estator, conforme mostrado na Figura 39.

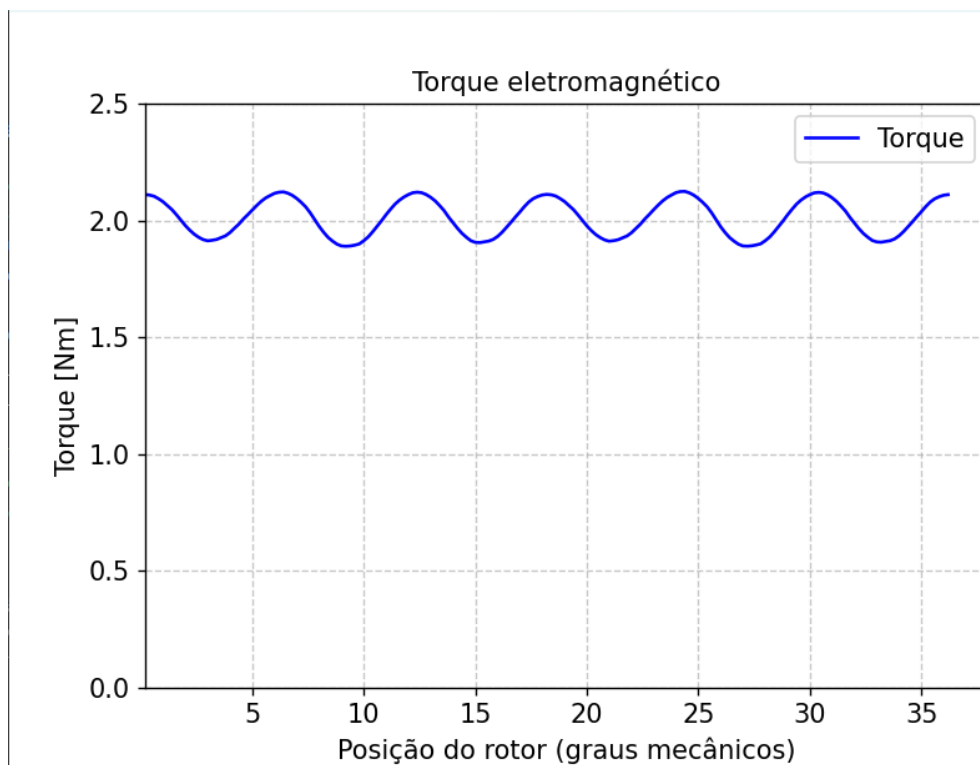
Figura 39 – Formas de onda das correntes de alimentação (fases A, B e C)



Fonte: Autora (2025).

A interação entre o campo magnético pelo estator e o campo do rotor resulta no torque eletromagnético, apresentado na Figura 40.

Figura 40 – Torque Eletromagnético



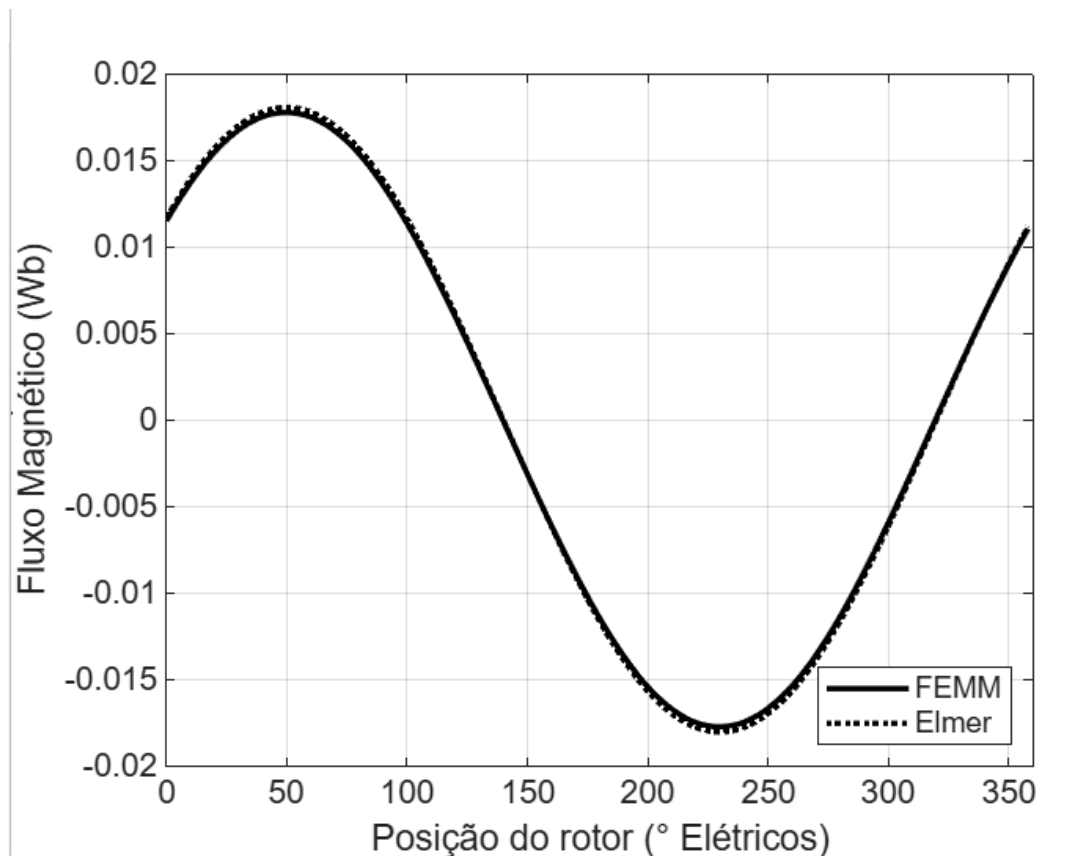
Fonte: Autora (2025).

4.2.3 Análise Comparativa e Validação dos Resultados

Para validar o modelo numérico, os resultados do Elmer foram confrontados com os dos cálculos de referência no FEMM, tanto para a condição a vazio quanto sob carga.

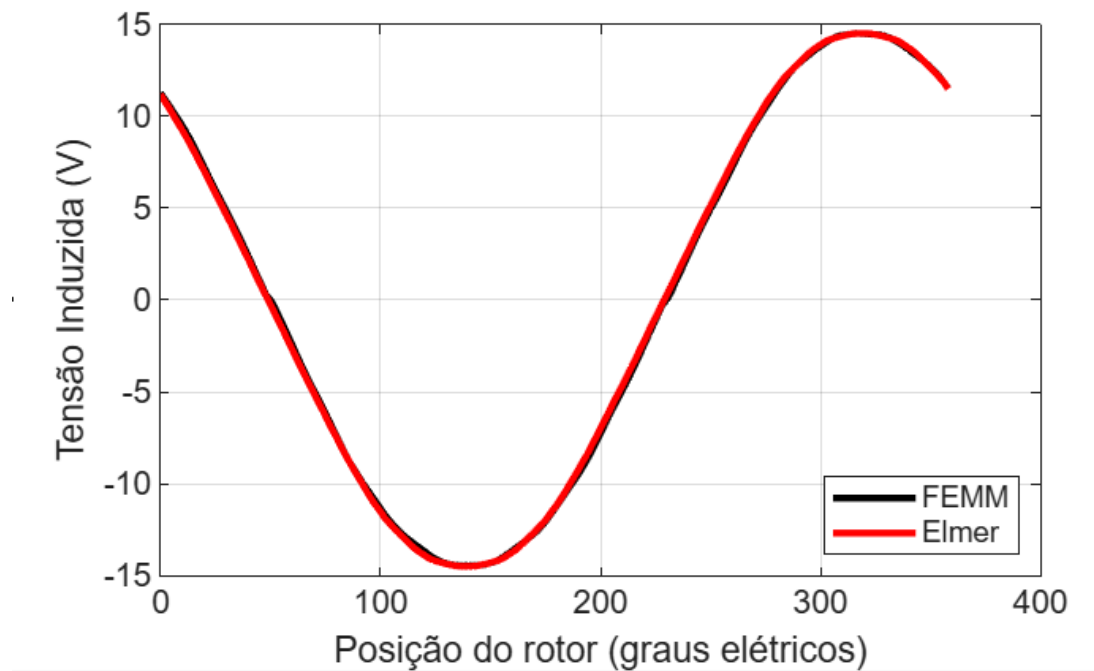
As Figuras 41 e 42 comparam, respectivamente, o fluxo concatenado e a tensão induzida. Em ambos os casos, observa-se uma concordância entre os modelos, tanto em amplitude quanto em forma de onda. Quantitativamente, essa concordância é confirmada pelos valores eficazes (*RMS*): o *RMS* do fluxo concatenado (rms_f) é de 0.0126 Wb no FEMM e 0.0128 Wb no Elmer, resultando em uma diferença de apenas 1.59%. Similarmente, o *RMS* da tensão induzida (rms_{emf}) é de 10.5254 V no FEMM e 10.5433 V no Elmer, com uma diferença percentual de apenas 0,17%.

Figura 41 – Comparação do fluxo magnético concatenado: Elmer vs. FEMM



Fonte: Autora (2025).

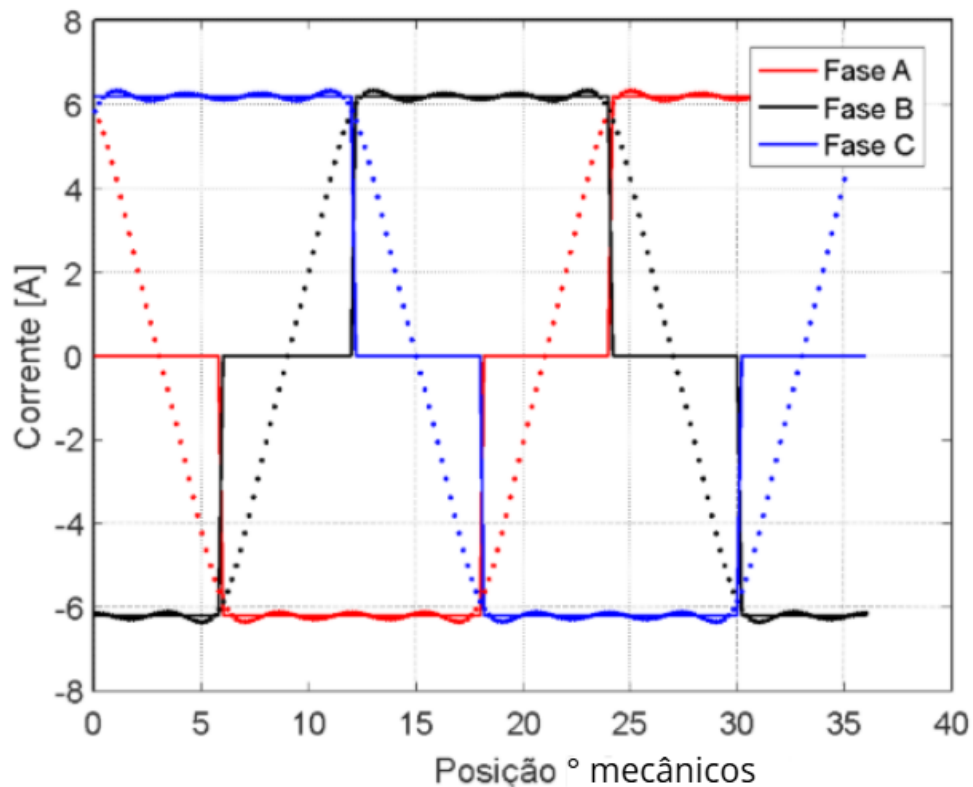
Figura 42 – Comparação da tensão induzida: Elmer vs. FEMM



Fonte: Autora (2025).

De forma análoga ao procedimento do Elmer, foi definido a alimentação do FEMM, conforme Figura 43. Nesse caso não foi feita a comparação entre as duas no mesmo gráfico, uma vez fica difícil a distinção de cada uma. Entretanto, esse resultado se aproxima das curvas das corrente do Elmer.

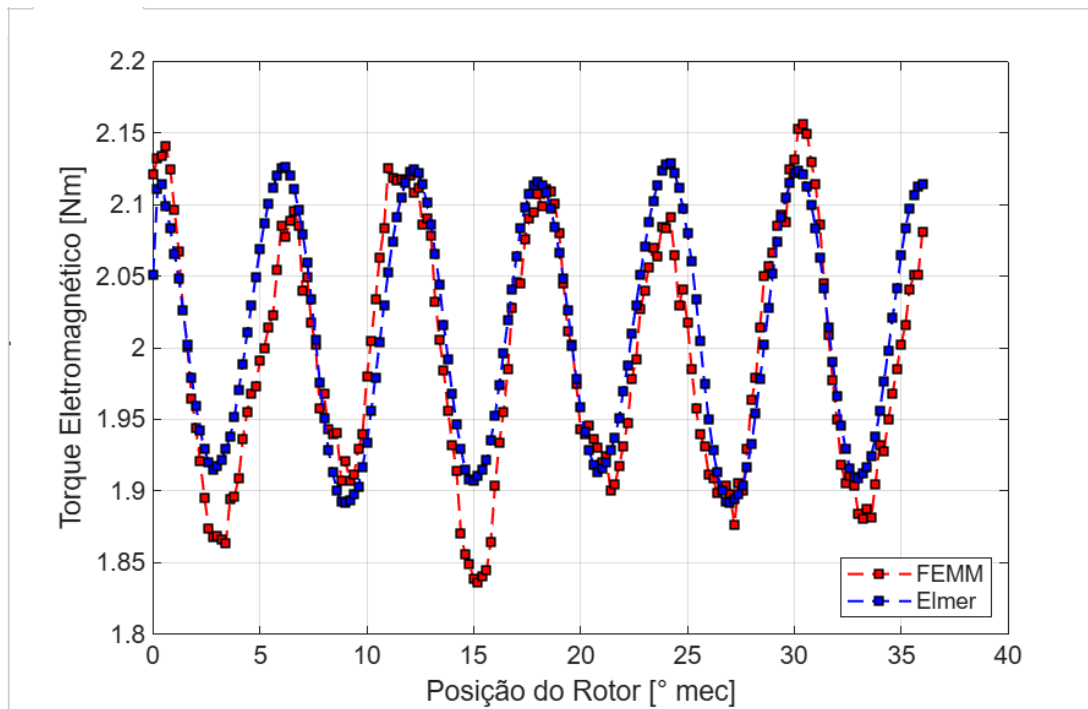
Figura 43 – Alimentação FEMM



Fonte: Autora (2025).

Por fim, a Figura 44 exibe a comparação do torque eletromagnético sob carga. Os resultados mostram uma similaridade, com valores médios e ondulação de torque próximos. Quantitativamente, o valor eficaz (*RMS*) do torque é de 1.9976 Nm no FEMM e 1.9471 Nm no Elmer, o que representa uma diferença percentual de 2.53%. Essa baixa divergência pode ser atribuída a diferenças nos algoritmos de cálculo de campo e de pós-processamento de cada software. No geral, a proximidade dos resultados valida a precisão do modelo implementado no Elmer para a análise do motor em estudo.

Figura 44 – Comparação do torque eletromagnético sob carga: Elmer vs. FEMM



Fonte: Autora (2025).

5 CONCLUSÃO

Este trabalho teve como objetivo principal a modelagem detalhada de um motor síncrono com ímãs permanentes no rotor externo, utilizando o software de código aberto Elmer. A motivação para este estudo reside na crescente necessidade de ferramentas de cálculos precisos e acessíveis para o projeto de máquinas elétricas eficientes, que são componentes chave na transição para uma economia de baixo carbono. Para atingir este objetivo, foram estabelecidas etapas claras, incluindo uma sucinta revisão teórica sobre o funcionamento de motores síncronos, um breve estudo geral sobre o método dos elementos finitos e a aplicação prática em uma cadeia de softwares livres.

A principal contribuição deste trabalho é a apresentação de um fluxo de trabalho, funcional e preciso para a análise de máquinas elétricas, baseado inteiramente em ferramentas de código aberto. Esta validação empírica do software Elmer como uma alternativa viável a pacotes comerciais representa um avanço significativo, democratizando o acesso a cálculos multifísicos avançados para estudantes, pesquisadores e pequenas empresas.

O detalhamento de cada etapa, desde a criação do script no Gmsh até o pós-processamento em Python, serve como um guia prático e um material didático para futuros trabalhos na área. Ao validar esta metodologia, o estudo não apenas alcança seus próprios objetivos, mas também capacita a comunidade acadêmica e técnica a desenvolver cálculos de motores, reduzindo barreiras financeiras e tecnológicas.

Apesar dos resultados positivos, reconhece-se que o presente estudo possui limitações que abrem caminhos para futuras investigações. Um estudo detalhado sobre a segregação de perdas na máquina deve ser feito, abordando detalhadamente os materiais. Além disso, a modelagem foi realizada em um domínio bidimensional (2D), que, embora forneça uma excelente aproximação, não captura efeitos tridimensionais (3D). Como sugestão para trabalhos futuros, recomenda-se a extensão deste modelo para uma análise 3D, o que permitiria um cálculo ainda mais preciso das perdas e do desempenho geral da máquina.

Outra evolução natural seria a validação experimental dos resultados, por meio da medição em bancada das grandezas calculadas. Adicionalmente, o caráter multifísico do Elmer pode ser explorado para acoplar uma análise térmica ao modelo eletromagnético, permitindo prever o comportamento da temperatura no motor.

REFERÊNCIAS

- BASTOS, J. P. A. **Eletromagnetismo para engenharia: Estática e Quase-Estática**. 2ª edição revista. ed. Florianópolis: Editora da UFSC, 2008.
- BIANCHI, N. **Electrical machine analysis using finite elements**. New York: Taylor and Francis, 2005.
- BOHRER, E. **Avaliação de materiais ferromagnéticos para um motor síncrono com rotor externo**. 2019. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) - Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis. 2019.
- DORNELLES, P. E. E. L. **Estudo de motores elétricos aplicados a veículos elétricos e híbridos**. 2020. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecânica) - Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis. 2020.
- FITZGERALD, A. E.; KINGSLEY, C.; UMANS, S. D. **Máquinas elétricas**. Porto Alegre: Bookman, 2006.
- GEUZAIN, C.; REMACLE, J.-F. **Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities**. 2009. <https://gmsh.info/>. Acesso em: 25 out. 2025.
- HADRAOUI, H. et al. A multi-criteria analysis and trends of electric motors for electric vehicles. **Applied System Innovation**, v. 3, n. 1, p. 10, 2020.
- HENDERSHOT, J. R.; MILLER, T. J. **Design of brushless permanent-magnet machines**. Oxford: Magna Physics Publishing, 1994.
- HENDERSHOT, J. R.; MILLER, T. J. **Design of brushless permanent-magnet machines**. 2. ed. Oxford: Magna Physics Publishing, 2010.
- JULIANI, A. D. P. **Análise do campo magnético de um motor de ímã permanente no rotor utilizando o método dos elementos finitos**. 2007. Dissertação (Mestrado em Engenharia Elétrica) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2007.
- JULIANI, A. D. P. **EMB-5628: Sistemas Motrizes II - Notas de Aula**. 2024. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/xmlui/bitstream/handle/riufcg/18957/THALES%20DE%20AGUIAR%20TAVARES%20-%20%20TCC%20ENG.%20ELÁTRICA%202018..pdf?sequence=1&isAllowed=y>. Acesso em: 15 dez. 2024.
- JULIANI, A. D. P.; GREFF, D. S.; VIDOR, D. L. R. Losses comparisons between external and internal rotors of permanent magnet synchronous motors. *In*: **Anais do XXV Congresso Brasileiro de Automática (CBA)**. Natal, RN: SBA - Sociedade Brasileira de Automática, 2024. p. 1–6. Acesso em: 2 dez. 2025. Disponível em: https://www.sba.org.br/open_journal_systems/index.php/cba/article/view/4253.

KARKI, A.; PHUYAL, S.; TULADHAR, D. Status of pure electric vehicle power train technology and future prospects. **Applied System Innovation**, 2020. Disponível em: <https://www.mdpi.com/2571-5577/3/3/35>. Acesso em: 15 dez. 2024.

MARKETSANDMARKETS. MarketsAndMarkets: News Reports, 2020. Future of E-Powertrain Market by Integration Type, Component, Propulsion, Vehicle Type and Region - Global Forecast to 2024 . Disponível em: <https://www.marketsandmarkets.com/new-reports.asp>. Acesso em: 11 out. 2024.

MEEKER, D. **Magnetics Tutorial**. 2018. <https://www.femm.info/wiki/magneticstutorial>. Acesso em: 18 out. 2025.

MEEKER, D. **Finite Element Method Magnetics (FEMM): User's Manual**. [S.l.], 2020. Disponível em: <http://www.femm.info/wiki/HomePage>.

ParaView. **Introdução ao ParaView**. [S.l.], 2025. Acesso em: 24 ago. 2025. Disponível em: <https://docs.paraview.org/en/latest/UsersGuide/introduction.html>.

PAULA, G. T. d. **Influência da saturação no torque da máquina síncrona de ímã permanente no rotor**. 2013. Dissertação (Dissertação de Mestrado em Engenharia Elétrica) — Universidade de São Paulo, Escola de Engenharia de São Carlos, São Carlos, SP, Brasil, 2013. Área de concentração: Sistemas Dinâmicos. Orientador: José Roberto Boffino de Almeida Monteiro. Disponível em: <https://teses.usp.br/teses/disponiveis/18/18153/tde-26082013-105401/pt-br.php>.

PINHEIRO, H. et al. Modulação space vector para inversores alimentados em tensão: uma abordagem unificada. **Revista Brasileira de Engenharia Elétrica**, 2005. Disponível em: <https://www.scielo.br/j/ca/a/vrQ7qBFjKW64y4tpbFHKjvr/>. Acesso em: 15 dez. 2024.

PONOMAREV, P. **FEM Modeling of PMSMs Using Elmer: Example workflow**. Lappeenranta, 2015. Disponível em: <https://www.researchgate.net/publication/275890336>.

RIMPAS, D. et al. Comparative review of motor technologies for electric vehicles powered by a hybrid energy storage system based on multi-criteria analysis. **Energies**, Roma, v. 16, n. 6, p. 2555, mar./abr. 2023.

RUOKOLAINEN, J. et al. **ElmerSolver Manual**. [S.l.], 2023. Manual de software, versão 9.0. Disponível em: <http://www.csc.fi/elmer>.

SANTOS, A. C. **Inversor Multinível com função de Filtro Ativo Paralelo para Sistemas Conectados à Rede Elétrica**. 2011. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal de Pernambuco, João Pessoa, 2011.

AGUIAR TAVARES, T. de. **Análise e Implementação de um Motor de Corrente Contínua Sem Escovas**. 2018. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) - Universidade Federal de Campina Grande, Campina Grande. 2018.

APÊNDICE A – CÓDIGO DE CÁLCULO DO TORQUE DE BORDA

```

1
2 Autor: Larissa Baqui o - 02/10/2025 - Adaptado relatorio Panvel
3 import pylab as pl
4 import numpy as np
5 import math
6
7 #abra o arquivo que voce quer ler
8 #Para dar menos erro de caminho, coloque seu codigo na mesma pasta
9 infile = open('scalars.dat', 'r')
10
11 # inicializacao do vetor para T
12 torque = []
13
14 # Define o comprimento do stack em metros (necessario para o calculo)
15 # IMPORTANTE: esse valor deve ser configurado de acordo com o modelo
16 stack_len = 0.036      #>>>>>>>>>>>>>>>>>>>>>>>>>CONFIGURAR!
17
18 # Loop para ler cada linha do arquivo
19 for line in infile:
20     line = line.split() # Divide cada linha em partes (separadas
    por espaco)
21     # o nome da variavel salva no .dat esta no .name: torque esta na
    coluna 2 (0,1,2)
22     torque.append(float(line[2])*stack_len)
23     #print(torque)
24
25 # Fecha o arquivo
26 infile.close()
27
28 # Plota a curva do torque
29 pl.figure()
30 pl.plot(torque,'o-')
31 pl.grid(True)
32 pl.title('Torque de Borda')
33 pl.xlabel('Posicao do rotor, [graus mecanicos]')
34 \ylabel('Torque, T [Nm]')
35 \pl.show()
```

Listing A.1 – Código para Cálculo do Torque de Borda

APÊNDICE B – CÓDIGO DE CÁLCULO DA TENSÃO INDUZIDA E FLUXO CONCATENADO

```

1 #LEMBRETE: PARA RODAR, DIGITAR F5
2 Autor: Larissa Baqui o - 02/10/2025 - Adaptado relatorio Panvel
3 import csv
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import os
7
8
9 # ler o arquivo com os dados
10 infile = open('./Tensao_indu.csv', 'r')
11
12 # inicia o vetor do fluxo
13 Tensao_indu = []
14
15 for line in infile:
16     line = line.split(',')
17     Tensao_indu.append(line[4]) # fourth word is Psi
18
19 # fecha o arquivo
20 infile.close()
21
22 # converte o vetor de string para float
23 Tensao_indu = np.array(Tensao_indu[1:], float)
24 print(Tensao_indu)
25
26 # calcula a tensao
27 #wm=n*2*pi/60;      w_el=wm*pp;      w_el=(n/60)*2*pi*pp
28 #Rotor externo: 800 rpm e 10 pares de polo
29
30
31 #(vel em rpm) e pp (pares de polos)
32 w_el = 800./60.*2.*np.pi*10.
33 U = (Tensao_indu[1:]-Tensao_indu[:-1])/(1./(w_el/2./np.pi)/180.)
34 print(U)
35
36 # plot curva da tesao induzida
37 plt.figure(1)
38 plt.plot(Tensao_indu, '-r')
39 plt.grid(True)
40 plt.title('Flux Linkage Curve')
41 plt.xlabel('Rotor position, [deg.mech]')
42 plt.ylabel('Flux linkage, [V.s]')

```

```

43 plt.show()
44 print(Tensao_indu)
45
46 #=====
47 #imprime em txt para usar no matlab - Fluxo
48 import numpy
49 Array = numpy.array(Tensao_indu)
50 x = len(Array)
51 print(x)
52
53 print('Array:\n', Array)
54 file = open("fluxo.txt", "w+")
55
56 content = str(Array)
57 file.write(content)
58 file.close()
59 #=====
60
61 plt.figure(2)
62 plt.plot(U, '-r', linewidth = 2.0)
63 plt.grid(True)
64 \title('Back-EMF Curve')
65 \xlabel('Rotor position, [deg.mech]')
66 \ylabel('Voltage, U [V]')
67 \plt.show()
68
69 #=====
70 #imprime em txt para usar no matlab - EMF
71 import numpy
72 Array_emf = numpy.array(U)
73 x1 = len(Array_emf)
74 print(x1)
75
76 print('Array_emf:\n', Array_emf)
77 file = open("emf.txt", "w+")
78
79 content = str(Array_emf)
80 file.write(content)
81 file.close()
82 #=====
83
84 # calcula harmonica
85 harm_U = np.abs( np.fft.fft(U) / len(U) ) * 2
86
87 # plota harmonicas
88 plt.figure(3)

```

Listing B.1 – Código para Cálculo da Tensão Induzida (Back-EMF)

APÊNDICE C – ARQUIVO .SIF PARA MODELAGEM 2D DO MOTOR - MOTOR SEM ALIMENTAÇÃO

```

1
2 Autor: Larissa Baquiao - 02/10/2025 - adaptado do relatorio Panvel
3 ! Rotor Externo - Calculo 2D Magnetoestatica
4 ! Parametros
5 $ w_m = (200/60)*2*pi ! [rad/s mech.] velocidade angular
6 $ pp = 10 ! numero de pares de polos
7 $ w_el = w_m*pp ! [rad/s el.] velocidade angular
8 $ B_PM = 1.17 ! [T] remanencia
9 $ mu_PM = 1.06 ! permeabilidade relativa dos imas
10 $ H_PM = B_PM/(mu_PM*pi*4d-7) ! [A/m] magnetizacao do imas
11 $ Scs = 0.0000415202 ! [m^2] area da ranhura
12
13
14 Header
15 CHECK KEYWORDS Warn
16 Mesh DB "rotor_ext" "."
17 Results Directory "results"
18 End
19
20 Constants
21 Permittivity of Vacuum = 8.8542e-12
22 End
23
24
25 !Gira no sentido anti horario
26 Simulation
27 Max Output Level = 3
28 Coordinate System = Cartesian
29 Coordinate Scaling = 0.001
30 Simulation Type = Transient
31 Timestepping Method = BDF
32 BDF Order = 2
33 Timestep Sizes = $ 1/(w_el/2/pi)/180! 180 amostras por periodo
    eletrico
34 Timestep Intervals = 181
35 Output Intervals = 1
36 Interpolation Numeric Epsilon = Real 5.0e-9
37 End
38
39
40 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Materiais !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
41 Material 1

```

```

42   Name = "Air"
43   Relative Permeability = 1
44 End
45
46 Material 2
47   Name = "Iron"
48   !Formato B-H
49   H-B Curve = Variable coupled iter
50   Real Monotone Cubic
51       0  0
52       0.10  74.7
53       0.20  97.5
54       0.30  110
55       0.40  120
56       0.50  130
57       0.60  140
58       0.70  150
59       0.80  162
60       0.90  175
61       1.00  190
62       1.10  208
63       1.20  227
64       1.30  265
65       1.40  366
66       1.50  633
67       1.60  1490
68       1.70  3670
69       1.80  7420
70       1.90  13000
71       2.0   21000
72       2.1   34000
73       2.2   55000
74       2.3   88000
75       2.4   140000
76       2.5   220000
77       2.6   349000
78       2.7   550000
79       2.8   860000
80       2.9   1350000
81       3.0  2200000
82       4.0  60000000
83   End
84
85
86 !Imas: posicao do ima inicial, magnetizacao d=dentro e f=fora
87 Material 3
88   !Magnetizacao 0d graus

```

```

89   Name = "0d"
90   Relative Permeability = $ mu_PM
91   Magnetization 1 = Variable time, timestep size
92     Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1)))"
93   Magnetization 2 = Variable time, timestep size
94     Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1)))"
95 End
96
97 Material 4
98   !Magnetizacao 18f graus
99   Name = "18f"
100  Relative Permeability = $ mu_PM
101  Magnetization 1 = Variable time, timestep size
102    Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(18*pi/180))"
103  Magnetization 2 = Variable time, timestep size
104    Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(18*pi/180))"
105 End
106
107 Material 5
108   !Magnetizacao 36d graus
109   Name = "36d"
110  Relative Permeability = $ mu_PM
111  Magnetization 1 = Variable time, timestep size
112    Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(36*pi/180))"
113  Magnetization 2 = Variable time, timestep size
114    Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(36*pi/180))"
115 End
116 Material 6
117   !Magnetizacao 54f graus
118   Name = "54f"
119  Relative Permeability = $ mu_PM
120  Magnetization 1 = Variable time, timestep size
121    Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(54*pi/180))"
122  Magnetization 2 = Variable time, timestep size
123    Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(54*pi/180))"
124 End
125 Material 7
126   !Magnetizacao 72d graus
127   Name = "72d"
128  Relative Permeability = $ mu_PM
129  Magnetization 1 = Variable time, timestep size
130    Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(72*pi/180))"
131  Magnetization 2 = Variable time, timestep size
132    Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(72*pi/180))"
133 End
134 Material 8
135   !Magnetizacao 90f graus

```

```

136 Name = "90f"
137 Relative Permeability = $ mu_PM
138 Magnetization 1 = Variable time, timestep size
139     Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1)))"
140 Magnetization 2 = Variable time, timestep size
141     Real MATC "H_PM*cos(w_m*(tx(0)-tx(1)))"
142 End
143 Material 9
144     !Magnetizacao 108d graus
145     Name = "108d"
146     Relative Permeability = $ mu_PM
147     Magnetization 1 = Variable time, timestep size
148         Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(108*pi/180))"
149     Magnetization 2 = Variable time, timestep size
150         Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(108*pi/180))"
151 End
152 Material 10
153     !Magnetizacao 126f graus
154     Name = "126f"
155     Relative Permeability = $ mu_PM
156     Magnetization 1 = Variable time, timestep size
157         Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(126*pi/180))"
158     Magnetization 2 = Variable time, timestep size
159         Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(126*pi/180))"
160 End
161 Material 11
162     !Magnetizacao 144d graus
163     Name = "144d"
164     Relative Permeability = $ mu_PM
165     Magnetization 1 = Variable time, timestep size
166         Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(144*pi/180))"
167     Magnetization 2 = Variable time, timestep size
168         Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(144*pi/180))"
169 End
170 Material 12
171     !Magnetizacao 162f graus
172     Name = "162f"
173     Relative Permeability = $ mu_PM
174     Magnetization 1 = Variable time, timestep size
175         Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(162*pi/180))"
176     Magnetization 2 = Variable time, timestep size
177         Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(162*pi/180))"
178 End
179 Material 13
180     !Magnetizacao 180d graus
181     Name = "180d"
182     Relative Permeability = $ mu_PM

```

```

183 Magnetization 1 = Variable time, timestep size
184 Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(180*pi/180))"
185 Magnetization 2 = Variable time, timestep size
186 Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(180*pi/180))"
187 End
188 Material 14
189 !Magnetizacao 198f graus
190 Name = "198f"
191 Relative Permeability = $ mu_PM
192 Magnetization 1 = Variable time, timestep size
193 Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(198*pi/180))"
194 Magnetization 2 = Variable time, timestep size
195 Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(198*pi/180))"
196 End
197 Material 15
198 !Magnetizacao 216d graus
199 Name = "216d"
200 Relative Permeability = $ mu_PM
201 Magnetization 1 = Variable time, timestep size
202 Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(216*pi/180))"
203 Magnetization 2 = Variable time, timestep size
204 Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(216*pi/180))"
205 End
206 Material 16
207 !Magnetizacao 234f graus
208 Name = "234f"
209 Relative Permeability = $ mu_PM
210 Magnetization 1 = Variable time, timestep size
211 Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(234*pi/180))"
212 Magnetization 2 = Variable time, timestep size
213 Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(234*pi/180))"
214 End
215 Material 17
216 !Magnetizacao 252d graus
217 Name = "252d"
218 Relative Permeability = $ mu_PM
219 Magnetization 1 = Variable time, timestep size
220 Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(252*pi/180))"
221 Magnetization 2 = Variable time, timestep size
222 Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(252*pi/180))"
223 End
224 Material 18
225 !Magnetizacao 270f graus
226 Name = "270f"
227 Relative Permeability = $ mu_PM
228 Magnetization 1 = Variable time, timestep size
229 Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(270*pi/180))"

```

```

230 Magnetization 2 = Variable time, timestep size
231 Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(270*pi/180))"
232 End
233 Material 19
234 !Magnetizacao 288d graus
235 Name = "288d"
236 Relative Permeability = $ mu_PM
237 Magnetization 1 = Variable time, timestep size
238 Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(288*pi/180))"
239 Magnetization 2 = Variable time, timestep size
240 Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(288*pi/180))"
241 End
242 Material 20
243 !Magnetizacao 306f graus
244 Name = "306f"
245 Relative Permeability = $ mu_PM
246 Magnetization 1 = Variable time, timestep size
247 Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(306*pi/180))"
248 Magnetization 2 = Variable time, timestep size
249 Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(306*pi/180))"
250 End
251 Material 21
252 !Magnetizacao 324d graus
253 Name = "324d"
254 Relative Permeability = $ mu_PM
255 Magnetization 1 = Variable time, timestep size
256 Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(324*pi/180))"
257 Magnetization 2 = Variable time, timestep size
258 Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(324*pi/180))"
259 End
260 Material 22
261 !Magnetizacao 342f graus
262 Name = "342f"
263 Relative Permeability = $ mu_PM
264 Magnetization 1 = Variable time, timestep size
265 Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(342*pi/180))"
266 Magnetization 2 = Variable time, timestep size
267 Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(342*pi/180))"
268 End
269
270 Material 23
271 Name = "Wedge"
272 Relative Permeability = 1
273 End
274
275 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Condicao de contorno
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

```

276 !outer boundary Dirichlet
277 Boundary Condition 1
278   Target Boundaries(1) = 1
279   Name = "Boundary_Outer"
280   Potential Condition = 0
281   Potential = 0
282   A = Real 0
283 End
284
285 Boundary Condition 2
286   Target Boundaries(1) = 2
287   Name = "Sliding"
288   Discontinuous Boundary = Logical True
289   Save Line = True
290   Mortar BC = 3
291   Anti Rotational Projector = Logical True
292   Galerkin Projector = Logical True
293 End
294
295 !!!!!!!!!!!!!!!!!!!!!!!!!!!!! Body Force!!!!!!!!!!!!!!!!!!!!!!!!!!!!
296 !rotacao do rotor
297 Body Force 1
298   Name = "BodyForce_Rotation"
299   Mesh Rotate 3 = Variable time, timestep size
300     Real MATC "180/pi*w_m*(tx(0)-tx(1))" ! in degrees
301 End
302
303 !!!!!!!!!!!!!!!!!!!!!!!!!!!!! Parametros dos Solvers
304   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!
305 !malha rotacao
306 Solver 1
307   Exec Solver = Before Timestep
308   Equation = MeshDeform
309   Procedure = "RigidMapper" "RigidMapper"
310 End
311
312 ! solver para vetor potencial magnetico A [Vs/m]
313 Solver 2
314   Equation = MgDyn2D
315   Procedure = "MagnetoDynamics2D" "MagnetoDynamics2D"
316   Exec Solver = Always
317
318   Handle Assembly = Logical True
319
320   Nonlinear System Convergence Tolerance = 1.0e-6
321   Nonlinear System Max Iterations = 20
322   Nonlinear System Relaxation Factor = 1

```

```

322
323   Linear System Solver = Direct
324   Linear System Direct Method = UMFPACK
325 End
326
327 !solver para inducao magnetica B [T]
328
329 Solver 3
330   Equation = CalcFields
331   Procedure = "Magnetodynamics" "MagnetodynamicsCalcFields"
332   Exec Solver = After Timestep
333
334   Linear System Solver = Direct
335   Linear System Direct Method = UMFPACK
336
337   Discontinuous Galerkin = True
338   Average Within Materials = False
339 End
340
341 !save arquivo no formato VTU para visualizacao no ParaView
342 Solver 4
343   Exec Solver = After Timestep
344   Procedure = "ResultOutputSolve" "ResultOutputSolver"
345   Output File Name = "step"
346   Vtu Format = True
347   Binary Output = True
348   Single Precision = True
349   Save Geometry Ids = True
350 End
351
352 !save variaveis escalares - torque T [Nm]
353 Solver 5
354   Exec Solver = After Timestep
355   Filename = "scalars.dat"
356   Procedure = "SaveData" "SaveScalars"
357   Show Norm Index = 1
358 End
359
360 Equation 1
361   Name = "ModelDomain"
362   Active Solvers(5) = 1 2 3 4 5
363 End
364
365 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
366
367 Body 1
368   Target Bodies(1) = 1

```

```

369   Name = "A+"
370   Equation = 1
371   Material = 1
372 End
373
374 Body 2
375   Target Bodies(1) = 2
376   Name = "A-"
377   Equation = 1
378   Material = 1
379 End
380
381 Body 3
382   Target Bodies(1) = 3
383   Name = "C+"
384   Equation = 1
385   Material = 1
386 End
387
388 Body 4
389   Target Bodies(1) = 4
390   Name = "C-"
391   Equation = 1
392   Material = 1
393 End
394
395 Body 5
396   Target Bodies(1) = 5
397   Name = "B+"
398   Equation = 1
399   Material = 1
400 End
401
402 Body 6
403   Target Bodies(1) = 6
404   Name = "B-"
405   Equation = 1
406   Material = 1
407 End
408
409 Body 7
410   Target Bodies(1) = 7
411   Name = "stator lamination"
412   Equation = 1
413   Material = 2
414 End
415

```

```

416 Body 8
417   Target Bodies(1) = 8
418   Name = "rotor lamination"
419   Equation = 1
420   Material = 2
421   Body Force = 1
422 End
423
424 Body 9
425   Target Bodies(1) = 9
426   Name = "wedges"
427   Equation = 1
428   Material = 23
429   Body Force = 1
430 End
431
432
433
434 Body 10
435   Target Bodies(1) = 12
436   Name = "PM1"
437   Equation = 1
438   Material = 3
439   Body Force = 1
440 End
441
442 Body 11
443   Target Bodies(1) = 13
444   Name = "PM2"
445   Equation = 1
446   Material = 4
447   Body Force = 1
448 End
449
450 Body 12
451   Target Bodies(1) = 14
452   Name = "PM3"
453   Equation = 1
454   Material = 5
455   Body Force = 1
456 End
457
458 Body 13
459   Target Bodies(1) = 15
460   Name = "PM4"
461   Equation = 1
462   Material = 6

```

```
463   Body Force = 1
464 End
465
466 Body 14
467   Target Bodies(1) = 16
468   Name = "PM5"
469   Equation = 1
470   Material = 7
471   Body Force = 1
472 End
473
474 Body 15
475   Target Bodies(1) = 17
476   Name = "PM6"
477   Equation = 1
478   Material = 8
479   Body Force = 1
480 End
481
482 Body 16
483   Target Bodies(1) = 18
484   Name = "PM7"
485   Equation = 1
486   Material = 9
487   Body Force = 1
488 End
489
490 Body 17
491   Target Bodies(1) = 19
492   Name = "PM8"
493   Equation = 1
494   Material = 10
495   Body Force = 1
496 End
497
498 Body 18
499   Target Bodies(1) = 20
500   Name = "PM9"
501   Equation = 1
502   Material = 11
503   Body Force = 1
504 End
505
506 Body 19
507   Target Bodies(1) = 21
508   Name = "PM10"
509   Equation = 1
```

```
510   Material = 12
511   Body Force = 1
512 End
513
514 Body 20
515   Target Bodies(1) = 22
516   Name = "PM11"
517   Equation = 1
518   Material = 13
519   Body Force = 1
520 End
521
522 Body 21
523   Target Bodies(1) = 23
524   Name = "PM12"
525   Equation = 1
526   Material = 14
527   Body Force = 1
528 End
529
530 Body 22
531   Target Bodies(1) = 24
532   Name = "PM13"
533   Equation = 1
534   Material = 15
535   Body Force = 1
536 End
537
538 Body 23
539   Target Bodies(1) = 25
540   Name = "PM14"
541   Equation = 1
542   Material = 16
543   Body Force = 1
544 End
545
546 Body 24
547   Target Bodies(1) = 26
548   Name = "PM15"
549   Equation = 1
550   Material = 17
551   Body Force = 1
552 End
553
554 Body 25
555   Target Bodies(1) = 27
556   Name = "PM16"
```

```

557 Equation = 1
558 Material = 18
559 Body Force = 1
560 End
561
562 Body 26
563 Target Bodies(1) = 28
564 Name = "PM17"
565 Equation = 1
566 Material = 19
567 Body Force = 1
568 End
569
570 Body 27
571 Target Bodies(1) = 29
572 Name = "PM18"
573 Equation = 1
574 Material = 20
575 Body Force = 1
576 End
577
578 Body 28
579 Target Bodies(1) = 30
580 Name = "PM19"
581 Equation = 1
582 Material = 21
583 Body Force = 1
584 End
585
586 Body 29
587 Target Bodies(1) = 31
588 Name = "PM20"
589 Equation = 1
590 Material = 22
591 Body Force = 1
592 End
593
594 Body 30
595 Target Bodies(1) = 10
596 Name = "stator airgap"
597 Equation = 1
598 Material = 1
599 End
600
601 Body 31
602 Target Bodies(1) = 11
603 Name = "rotor airgap"

```

```
604 Equation = 1
605 Material = 1
606 Body Force = 1
607 r outer = Real 0.04550
608 r inner = Real 0.04475
609 End
```

Listing C.1 – Código para Cálculo da Tensão Induzida

APÊNDICE D – ARQUIVO .SIF PARA CÁLCULO DO MOTOR COM ALIMENTAÇÃO

```

1
2 ! Rotor Externo - Calculo 2D Magnetostatica
3 ! Parametros
4 Autor: Larissa Baqui o - 02/10/2025 - Adaptado relatorio Panvel
5 $ rpm = 200 ! velocidade de rotacao (1/min)
6 $ pp = 10 ! numero de pares de polos
7 $ f = rpm/60*pp ! frequencia eletrica (Hz)
8 $ w_m = (rpm/60)*2*pi ! [rad/s mech.] velocidade angular
9 $ w_el = w_m*pp ! [rad/s el.] velocidade angular eletrica
10 $ B_PM = 1.17 ! [T] remanencia dos imas
11 $ mu_PM = 1.06 ! permeabilidade relativa dos imas
12 $ H_PM = B_PM/(mu_PM*pi*4d-7) ! [A/m] magnetizacao dos imas permanentes
13 $ Nph = 10 ! Numero de espiras por fases
14 $ pico = 6.2 ! corrente de pico
15 $ Scs = 0.0000415202 ! [m^2] area da ranhura
16
17
18 Header
19   CHECK KEYWORDS Warn
20   Mesh DB "rotor_ext" "."
21   Results Directory "results"
22 End
23
24 Constants
25   Permittivity of Vacuum = 8.8542e-12
26 End
27
28
29 !Gira no sentido anti horario
30 Simulation
31   Max Output Level = 3
32   Coordinate System = Cartesian
33   Coordinate Scaling = 0.001
34   Simulation Type = Transient
35   Timestepping Method = BDF
36   BDF Order = 2
37   Timestep Sizes = $ 1/(w_el/2/pi)/180! 180 amostras por periodo
        eletrico
38   Timestep Intervals = 181
39   Output Intervals = 1
40   Interpolation Numeric Epsilon = Real 5.0e-9
41 End

```

```

42
43 !!!!!!!!!!!!!!!!!!!!!!!!!!!!! Materials !!!!!!!!!!!!!!!!!!!!!!!!!!!!!
44 Material 1
45   Name = "Air"
46   Relative Permeability = 1
47   Relative Permittivity = 1
48   Electric Conductivity = 0
49 End
50
51 Material 2
52   Name = "Iron"
53   H-B Curve = Variable coupled iter
54   Real Monotone Cubic
55   0 0
56     0.10  33.3
57     0.20  43.9
58     0.30  51.2
59     0.40  57.7
60     0.50  64.2
61     0.60  71.2
62     0.70  79.1
63     0.80  88.4
64     0.90  100.0
65     1.00  116.0
66     1.10  175.0
67     1.30  251.0
68     1.40  470.0
69     1.50  1230.0
70     1.60  3070.0
71     1.70  6150.0
72     1.80  10700.0
73     !Include BH
74   End
75
76
77 !Imas: posicao do ima inicial, magnetizacao d=dentro e f=fora
78 Material 3
79   !Magnetizacao 0d graus
80   Name = "0d"
81   Relative Permeability = $ mu_PM
82   Magnetization 1 = Variable time, timestep size
83     Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1)))"
84   Magnetization 2 = Variable time, timestep size
85     Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1)))"
86 End
87
88 Material 4

```

```

89  !Magnetizacao 18f graus
90  Name = "18f"
91  Relative Permeability = $ mu_PM
92  Magnetization 1 = Variable time, timestep size
93      Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(18*pi/180))"
94  Magnetization 2 = Variable time, timestep size
95      Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(18*pi/180))"
96 End
97
98 Material 5
99  !Magnetizacao 36d graus
100 Name = "36d"
101 Relative Permeability = $ mu_PM
102 Magnetization 1 = Variable time, timestep size
103     Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(36*pi/180))"
104 Magnetization 2 = Variable time, timestep size
105     Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(36*pi/180))"
106 End
107 Material 6
108  !Magnetizacao 54f graus
109 Name = "54f"
110 Relative Permeability = $ mu_PM
111 Magnetization 1 = Variable time, timestep size
112     Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(54*pi/180))"
113 Magnetization 2 = Variable time, timestep size
114     Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(54*pi/180))"
115 End
116 Material 7
117  !Magnetizacao 72d graus
118 Name = "72d"
119 Relative Permeability = $ mu_PM
120 Magnetization 1 = Variable time, timestep size
121     Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(72*pi/180))"
122 Magnetization 2 = Variable time, timestep size
123     Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(72*pi/180))"
124 End
125 Material 8
126  !Magnetizacao 90f graus
127 Name = "90f"
128 Relative Permeability = $ mu_PM
129 Magnetization 1 = Variable time, timestep size
130     Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1)))"
131 Magnetization 2 = Variable time, timestep size
132     Real MATC "H_PM*cos(w_m*(tx(0)-tx(1)))"
133 End
134 Material 9
135  !Magnetizacao 108d graus

```

```

136 Name = "108d"
137 Relative Permeability = $ mu_PM
138 Magnetization 1 = Variable time, timestep size
139     Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(108*pi/180))"
140 Magnetization 2 = Variable time, timestep size
141     Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(108*pi/180))"
142 End
143 Material 10
144     !Magnetizacao 126f graus
145     Name = "126f"
146     Relative Permeability = $ mu_PM
147     Magnetization 1 = Variable time, timestep size
148         Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(126*pi/180))"
149     Magnetization 2 = Variable time, timestep size
150         Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(126*pi/180))"
151 End
152 Material 11
153     !Magnetizacao 144d graus
154     Name = "144d"
155     Relative Permeability = $ mu_PM
156     Magnetization 1 = Variable time, timestep size
157         Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(144*pi/180))"
158     Magnetization 2 = Variable time, timestep size
159         Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(144*pi/180))"
160 End
161 Material 12
162     !Magnetizacao 162f graus
163     Name = "162f"
164     Relative Permeability = $ mu_PM
165     Magnetization 1 = Variable time, timestep size
166         Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(162*pi/180))"
167     Magnetization 2 = Variable time, timestep size
168         Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(162*pi/180))"
169 End
170 Material 13
171     !Magnetizacao 180d graus
172     Name = "180d"
173     Relative Permeability = $ mu_PM
174     Magnetization 1 = Variable time, timestep size
175         Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(180*pi/180))"
176     Magnetization 2 = Variable time, timestep size
177         Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(180*pi/180))"
178 End
179 Material 14
180     !Magnetizacao 198f graus
181     Name = "198f"
182     Relative Permeability = $ mu_PM

```

```

183 Magnetization 1 = Variable time, timestep size
184 Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(198*pi/180))"
185 Magnetization 2 = Variable time, timestep size
186 Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(198*pi/180))"
187 End
188 Material 15
189 !Magnetizacao 216d graus
190 Name = "216d"
191 Relative Permeability = $ mu_PM
192 Magnetization 1 = Variable time, timestep size
193 Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(216*pi/180))"
194 Magnetization 2 = Variable time, timestep size
195 Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(216*pi/180))"
196 End
197 Material 16
198 !Magnetizacao 234f graus
199 Name = "234f"
200 Relative Permeability = $ mu_PM
201 Magnetization 1 = Variable time, timestep size
202 Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(234*pi/180))"
203 Magnetization 2 = Variable time, timestep size
204 Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(234*pi/180))"
205 End
206 Material 17
207 !Magnetizacao 252d graus
208 Name = "252d"
209 Relative Permeability = $ mu_PM
210 Magnetization 1 = Variable time, timestep size
211 Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(252*pi/180))"
212 Magnetization 2 = Variable time, timestep size
213 Real MATC "-H_PM*sin(w_m*(tx(0)-tx(1))+(252*pi/180))"
214 End
215 Material 18
216 !Magnetizacao 270f graus
217 Name = "270f"
218 Relative Permeability = $ mu_PM
219 Magnetization 1 = Variable time, timestep size
220 Real MATC "H_PM*cos(w_m*(tx(0)-tx(1))+(270*pi/180))"
221 Magnetization 2 = Variable time, timestep size
222 Real MATC "H_PM*sin(w_m*(tx(0)-tx(1))+(270*pi/180))"
223 End
224 Material 19
225 !Magnetizacao 288d graus
226 Name = "288d"
227 Relative Permeability = $ mu_PM
228 Magnetization 1 = Variable time, timestep size
229 Real MATC "-H_PM*cos(w_m*(tx(0)-tx(1))+(288*pi/180))"

```



```

276 Target Boundaries(1) = 1
277 Name = "Boundary_Outer"
278 Potential Condition = 0
279 Potential = 0
280 A = Real 0
281 End
282
283 Boundary Condition 2
284 Target Boundaries(1) = 2
285 Name = "Sliding"
286 Discontinuous Boundary = Logical True
287 Save Line = True
288 Mortar BC = 3
289 Anti Rotational Projector = Logical True
290 Galerkin Projector = Logical True
291 End
292
293 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Body Force !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
294 !rotation of the rotor
295 Body Force 1
296 Name = "BodyForce_Rotation"
297 Mesh Rotate 3 = Variable time, timestep size
298 Real MATC "180/pi*w_m*(tx(0)-tx(1))+2" ! rotacionado em 2 graus para
    coincidir com a alimentacao do FEMM
299 End
300
301 !-----
302 !Funcao densidade de corrente
303 !-----
304
305 $ function Jap(tx) import Nph, pico, Scs,w_el { _Jap = (pico*Nph/Scs)
    *((0.608*cos(w_el*(tx(0)-tx(1))))+(0.27*cos(3*w_el*(tx(0)-tx(1))))
    +(0.024*cos(5*w_el*(tx(0)-tx(1))))+(0.012*cos(7*w_el*(tx(0)-tx(1))))
    +(0.03*cos(9*w_el*(tx(0)-tx(1))))-(1.053*sin(w_el*(tx(0)-tx(1))))
    +(0.042*sin(5*w_el*(tx(0)-tx(1))))-(0.021*sin(7*w_el*(tx(0)-tx(1))))
    ; }
306
307 $ function Jan(tx) import Nph, pico, Scs,w_el { _Jan = -(pico*Nph/Scs)
    *((0.608*cos(w_el*(tx(0)-tx(1))))+(0.27*cos(3*w_el*(tx(0)-tx(1))))
    +(0.024*cos(5*w_el*(tx(0)-tx(1))))+(0.012*cos(7*w_el*(tx(0)-tx(1))))
    +(0.03*cos(9*w_el*(tx(0)-tx(1))))-(1.053*sin(w_el*(tx(0)-tx(1))))
    +(0.042*sin(5*w_el*(tx(0)-tx(1))))-(0.021*sin(7*w_el*(tx(0)-tx(1))))
    ; }
308
309 $ function Jbp(tx) import Nph, pico, Scs,w_el { _Jbp = (pico*Nph/Scs)
    *(-(1.216*cos(w_el*(tx(0)-tx(1))))+(0.27*cos(3*w_el*(tx(0)-tx(1))))
    -(0.049*cos(5*w_el*(tx(0)-tx(1))))-(0.025*cos(7*w_el*(tx(0)-tx(1))))

```

```

+ (0.03*cos(9*w_el*(tx(0)-tx(1))))); }
310
311 $ function Jbn(tx) import Nph, pico, Scs, w_el { _Jbn = -(pico*Nph/Scs)
* (- (1.216*cos(w_el*(tx(0)-tx(1)))) + (0.27*cos(3*w_el*(tx(0)-tx(1))))
- (0.049*cos(5*w_el*(tx(0)-tx(1)))) - (0.025*cos(7*w_el*(tx(0)-tx(1))))
+ (0.03*cos(9*w_el*(tx(0)-tx(1))))); }
312
313 $ function Jcp(tx) import Nph, pico, Scs, w_el { _Jcp = (pico*Nph/Scs)
* ((0.608*cos(w_el*(tx(0)-tx(1)))) + (0.27*cos(3*w_el*(tx(0)-tx(1))))
+ (0.024*cos(5*w_el*(tx(0)-tx(1)))) + (0.012*cos(7*w_el*(tx(0)-tx(1))))
+ (0.03*cos(9*w_el*(tx(0)-tx(1)))) + (1.053*sin(w_el*(tx(0)-tx(1))))
- (0.042*sin(5*w_el*(tx(0)-tx(1)))) + (0.021*sin(7*w_el*(tx(0)-tx(1))))
; }
314
315 $ function Jcn(tx) import Nph, pico, Scs, w_el { _Jcn = -(pico*Nph/Scs)
* ((0.608*cos(w_el*(tx(0)-tx(1)))) + (0.27*cos(3*w_el*(tx(0)-tx(1))))
+ (0.024*cos(5*w_el*(tx(0)-tx(1)))) + (0.012*cos(7*w_el*(tx(0)-tx(1))))
+ (0.03*cos(9*w_el*(tx(0)-tx(1)))) + (1.053*sin(w_el*(tx(0)-tx(1))))
- (0.042*sin(5*w_el*(tx(0)-tx(1)))) + (0.021*sin(7*w_el*(tx(0)-tx(1))))
; }
316
317
318 !-----
319 ! Funcoes de Corrente por Fase (para salvar no arquivo)
320 !-----
321 $ function I_A(tx) import pico, w_el { _I_A = pico * ((0.608*cos(w_el*(
tx(0)-tx(1)))) + (0.27*cos(3*w_el*(tx(0)-tx(1)))) + (0.024*cos(5*w_el*(tx
(0)-tx(1)))) + (0.012*cos(7*w_el*(tx(0)-tx(1)))) + (0.03*cos(9*w_el*(tx
(0)-tx(1)))) - (1.053*sin(w_el*(tx(0)-tx(1)))) + (0.042*sin(5*w_el*(tx(0)
-tx(1)))) - (0.021*sin(7*w_el*(tx(0)-tx(1))))); }
322
323 $ function I_B(tx) import pico, w_el { _I_B = pico * (- (1.216*cos(w_el*(
tx(0)-tx(1)))) + (0.27*cos(3*w_el*(tx(0)-tx(1)))) - (0.049*cos(5*w_el*(tx
(0)-tx(1)))) - (0.025*cos(7*w_el*(tx(0)-tx(1)))) + (0.03*cos(9*w_el*(tx
(0)-tx(1))))); }
324
325 $ function I_C(tx) import pico, w_el { _I_C = pico * ((0.608*cos(w_el*(
tx(0)-tx(1)))) + (0.27*cos(3*w_el*(tx(0)-tx(1)))) + (0.024*cos(5*w_el*(tx
(0)-tx(1)))) + (0.012*cos(7*w_el*(tx(0)-tx(1)))) + (0.03*cos(9*w_el*(tx
(0)-tx(1)))) + (1.053*sin(w_el*(tx(0)-tx(1)))) - (0.042*sin(5*w_el*(tx(0)
-tx(1)))) + (0.021*sin(7*w_el*(tx(0)-tx(1))))); }
326
327 $ function TesteSimples (tx) { _TesteSimples = 10.0 * tx(0); }
328
329 !current in the phase A+
330 Body Force 2
331 Name = "Current_A_plus"

```

```

332 Current Density = Variable time, timestep size; Real MATC "Jap(tx)"
333 Joule Heat = Logical True
334 End
335
336 !current in the phase A-
337 Body Force 3
338 Name = "Current_A_minus"
339 Current Density = Variable time, timestep size; Real MATC "Jan(tx)"
340 Joule Heat = Logical True
341 End
342
343 !current in the phase B+
344 Body Force 4
345 Name = "Current_B_plus"
346 Current Density = Variable time, timestep size; Real MATC "Jbp(tx)"
347 Joule Heat = Logical True
348 End
349
350 !current in the phase B-
351 Body Force 5
352 Name = "Current_B_minus"
353 Current Density = Variable time, timestep size; Real MATC "Jbn(tx)"
354 Joule Heat = Logical True
355 End
356
357 !current in the phase C+
358 Body Force 6
359 Name = "Current_C_plus"
360 Current Density = Variable time, timestep size; Real MATC "Jcp(tx)"
361 Joule Heat = Logical True
362 End
363
364 !current in the phase C-
365 Body Force 7
366 Name = "Current_C_minus"
367 Current Density = Variable time, timestep size; Real MATC "Jcn(tx)"
368 Joule Heat = Logical True
369 End
370
371
372
373 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Parametros dos Solvers
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
374 !rotacao da malha
375 Solver 1
376 Exec Solver = Before Timestep
377 Equation = MeshDeform

```

```

378 Procedure = "RigidMeshMapper" "RigidMeshMapper"
379 End
380
381 ! Solver para o vetor do potencial magnetico A [Vs/m]
382 Solver 2
383 Equation = MgDyn2D
384 Procedure = "Magnetodynamics2D" "Magnetodynamics2D"
385 Exec Solver = Always
386 Variable = A
387 Handle Assembly = Logical True
388
389 Nonlinear System Convergence Tolerance = 1.0e-6
390 Nonlinear System Max Iterations = 20
391 Nonlinear System Relaxation Factor = 1
392
393 Linear System Solver = Direct
394 Linear System Direct Method = UMFPACK
395 Calculate Joule Heating = Logical True
396 End
397
398 !solver para a densidade de fluxo magnetico B [T]
399 Solver 3
400 Equation = CalcFields
401 Potential Variable = "A"
402 Procedure = "Magnetodynamics" "MagnetodynamicsCalcFields"
403 Linear System Solver = Direct
404 Exec Solver = Always
405 Calculate Nodal Forces = Logical True
406 Calculate Magnetic Vector Potential = Logical True
407 Calculate Winding Voltage = Logical True
408 Calculate Current Density = Logical True
409 Calculate Joule Heating = Logical True
410 Calculate Maxwell Stress = Logical True
411 Calculate JxB = Logical True
412 Calculate Magnetic Field Strength = Logical True
413
414 End
415
416
417 !Salva arquivo no formato VTU para visualizacao no ParaView
418 Solver 4
419 Exec Solver = After Timestep
420 Procedure = "ResultOutputSolve" "ResultOutputSolver"
421 Output File Name = "step"
422 Vtu Format = True
423 Binary Output = True
424 Single Precision = True

```

[illegible]

```

alimentacao
472 End
473
474 Body 2
475     Target Bodies(1) = 2
476     Name = "A-"
477     Equation = 1
478     Material = 24
479     Body Force = 3          !<<<<<<<<<<<<<<<<<<<<<<<<<<<<inserir de acordo com a
        alimentacao
480 End
481
482 Body 3
483     Target Bodies(1) = 3
484     Name = "C+"
485     Equation = 1
486     Material = 24
487     Body Force = 6          !<<<<<<<<<<<<<<<<<<<<<<<<<<<<inserir de acordo com a
        alimentacao
488 End
489
490 Body 4
491     Target Bodies(1) = 4
492     Name = "C-"
493     Equation = 1
494     Material = 24
495     Body Force = 7          !<<<<<<<<<<<<<<<<<<<<<<<<<<<<inserir de acordo com a
        alimentacao
496 End
497
498 Body 5
499     Target Bodies(1) = 5
500     Name = "B+"
501     Equation = 1
502     Material = 24
503     Body Force = 4          !<<<<<<<<<<<<<<<<<<<<<<<<<<<<inserir de acordo com a
        alimentacao
504 End
505
506 Body 6
507     Target Bodies(1) = 6
508     Name = "B-"
509     Equation = 1
510     Material = 24
511     Body Force = 5          !<<<<<<<<<<<<<<<<<<<<<<<<<<<<inserir de acordo com a
        alimentacao
512 End

```

```

513
514 Body 7
515     Target Bodies(1) = 7
516     Name = "stator lamination"
517     Equation = 1
518     Material = 2
519 End
520
521 Body 8
522     Target Bodies(1) = 8
523     Name = "rotor lamination"
524     Equation = 1
525     Material = 2
526     Body Force = 1
527 End
528
529 Body 9
530     Target Bodies(1) = 9
531     Name = "wedges"
532     Equation = 1
533     Material = 23
534     Body Force = 1
535 End
536
537 Body 10
538     Target Bodies(1) = 12
539     Name = "PM1"
540     Equation = 1
541     Material = 3
542     Body Force = 1
543 End
544
545 Body 11
546     Target Bodies(1) = 13
547     Name = "PM2"
548     Equation = 1
549     Material = 4
550     Body Force = 1
551 End
552
553 Body 12
554     Target Bodies(1) = 14
555     Name = "PM3"
556     Equation = 1
557     Material = 5
558     Body Force = 1
559 End

```

```
560
561 Body 13
562   Target Bodies(1) = 15
563   Name = "PM4"
564   Equation = 1
565   Material = 6
566   Body Force = 1
567 End
568
569 Body 14
570   Target Bodies(1) = 16
571   Name = "PM5"
572   Equation = 1
573   Material = 7
574   Body Force = 1
575 End
576
577 Body 15
578   Target Bodies(1) = 17
579   Name = "PM6"
580   Equation = 1
581   Material = 8
582   Body Force = 1
583 End
584
585 Body 16
586   Target Bodies(1) = 18
587   Name = "PM7"
588   Equation = 1
589   Material = 9
590   Body Force = 1
591 End
592
593 Body 17
594   Target Bodies(1) = 19
595   Name = "PM8"
596   Equation = 1
597   Material = 10
598   Body Force = 1
599 End
600
601 Body 18
602   Target Bodies(1) = 20
603   Name = "PM9"
604   Equation = 1
605   Material = 11
606   Body Force = 1
```

```
607 End
608
609 Body 19
610   Target Bodies(1) = 21
611   Name = "PM10"
612   Equation = 1
613   Material = 12
614   Body Force = 1
615 End
616
617 Body 20
618   Target Bodies(1) = 22
619   Name = "PM11"
620   Equation = 1
621   Material = 13
622   Body Force = 1
623 End
624
625 Body 21
626   Target Bodies(1) = 23
627   Name = "PM12"
628   Equation = 1
629   Material = 14
630   Body Force = 1
631 End
632
633 Body 22
634   Target Bodies(1) = 24
635   Name = "PM13"
636   Equation = 1
637   Material = 15
638   Body Force = 1
639 End
640
641 Body 23
642   Target Bodies(1) = 25
643   Name = "PM14"
644   Equation = 1
645   Material = 16
646   Body Force = 1
647 End
648
649 Body 24
650   Target Bodies(1) = 26
651   Name = "PM15"
652   Equation = 1
653   Material = 17
```

```
654   Body Force = 1
655 End
656
657 Body 25
658   Target Bodies(1) = 27
659   Name = "PM16"
660   Equation = 1
661   Material = 18
662   Body Force = 1
663 End
664
665 Body 26
666   Target Bodies(1) = 28
667   Name = "PM17"
668   Equation = 1
669   Material = 19
670   Body Force = 1
671 End
672
673 Body 27
674   Target Bodies(1) = 29
675   Name = "PM18"
676   Equation = 1
677   Material = 20
678   Body Force = 1
679 End
680
681 Body 28
682   Target Bodies(1) = 30
683   Name = "PM19"
684   Equation = 1
685   Material = 21
686   Body Force = 1
687 End
688
689 Body 29
690   Target Bodies(1) = 31
691   Name = "PM20"
692   Equation = 1
693   Material = 22
694   Body Force = 1
695 End
696
697 Body 30
698   Target Bodies(1) = 10
699   Name = "stator airgap"
700   Equation = 1
```

```
701   Material = 1
702 End
703
704 Body 31
705   Target Bodies(1) = 11
706   Name = "rotor airgap"
707   Equation = 1
708   Material = 1
709   Body Force = 1
710   r outer = Real 0.04550
711   r inner = Real 0.04475
712 End
```

Listing D.1 – Código para Cálculo do motor com alimentação